# Computer Chronicles Revisited

## Volume 1 (1983 - 1984)

S.M. Oliva

# Computer Chronicles Revisited, Volume 1

## 1983 - 1984

S.M. Oliva

S.M. Oliva, LLC

2022-06-06

© 2022 S.M. Oliva, LLC, CC BY-NC 4.0

# Introduction

This ebook reprints a series of posts from my blog *Computer Chronicles Revisited*, which I launched in May 2021. Each chapter covers a single episode of the PBS series *Computer Chronicles* that aired during that show's first season (1983-1984). There is an extended recap of each episode followed by analysis of what became of the people and products featured on the program. Each chapter then concludes with a miscellany section called "Notes from the Random Access File," an homage to the news segment that followed most *Computer Chronicles* broadcasts.

As of this writing in June 2022, I have covered the first two seasons of *Computer Chronicles* and am completing work on the third season. If you are interested in supporting my work, please consider joining my Patreon or visiting my Support page.

# Chapter 1: The HP-150 Touchscreen

*The Computer Chronicles* debuted as a national television program in the United States in the fall of 1983. The series was the brainchild of Stewart Cheifet, then the general manager at KCSM-TV, a public television station based in San Mateo, California, located in the heart of Silicon Valley. Cheifet originally launched *Chronicles* as a live, local program in 1981, which was hosted by Jim Warren, the co-founder of the West Coast Computer Faire, an annual computer convention based in San Francisco.

Cheifet told host Leo Laporte in a 2013 episode of the podcast Triangulation that Warren's local show–essentially a televised computer user group–went viral before the term "viral" existed. Other PBS stations around the country started calling Cheifet asking if they could carry the show. Eventually, Cheifet decided to relaunch *The Computer Chronicles* as a taped program with himself as the host and went looking for underwriters.

One potential sponsor Cheifet spoke with was Gary Kildall, the founder of Digital Research, Inc., which developed the popular CP/M operating system. Not only was Kildall "totally supportive" of the idea, Cheifet told Laporte, but he offered to co-host the program. Cheifet noted that Kildall was an experienced pilot and often flew his private plane up to San Mateo each Saturday to record.

*The Computer Chronicles* went on to run for approximately two decades, from 1984 until 2003. Cheifet served as principal host for the entire run and later created a companion program, *Net Cafe*, which focused on the growth of the Internet during the 1990s. Kildall would continue as co-host of *Chronicles*, with intermittent absences, until around 1990.

## What Was *The Computer Chronicles*?

*The Computer Chronicles* was essentially a newsmagazine-style program. Each show typically began with a brief host segment where Cheifet and Kildall would discuss the subject for the week. This led into a pre-recorded feature narrated by Cheifet or another contributor. There would then be one or two segments involving product demonstrations. A manager or developer from a given company would talk to Cheifet and Kildall in-studio and show off their latest project, which related back to the subject of the week.

After the first season, broadcast during 1984, *Chronicles* would follow the product demo segment with a brief roundup of technology-related headlines called “Random Access,” again presented by either Cheifet or a journalist affiliated with WITF-TV in Harrisburg, which co-produced the program nationally for PBS. This segment sometimes featured a brief, one- or two-minute software review.

## A Nostalgic Look Back at Computing History…in 1983

The subject of the first nationally broadcast episode was “Mainframes to Minis to Micros.” Cheifet opened the host segment by asking Kildall, presumably with tongue firmly planted in cheek, if he thought we were “at the end of the line” with respect to the evolution of computers. Obviously, Kildall replied no, pointing to the new trend of using microprocessors as general purpose computers. He said eventually, computers would get so small you would “lose them like your keys.” He even saw microprocessors being used as controllers for non-computing devices such as wheelchairs.

This led into the episode’s pre-recorded feature, where Cheifet narrated an introduction to the Computer Museum of Boston, a nonprofit organization that housed a collection of many early computers and calculating devices, such as the TX-0, the Pascaline, the Illiac IV, and the Whirlwind I.

The feature continued into a second segment, which included snippets from an interview with C. Gordon Bell, a member of the Museum’s board of directors and a former vice president at Digital Equipment Corporation, where he worked on the TX-0. The segment then transitioned back to the studio, where Cheifet and Kildall spoke with Herbert Lechner, vice president for information systems and administration at SRI International, a nonprofit scientific research institute originally founded by Stanford University. Cheiefet asked Lechner if he believed that micro computers–i.e., personal computers–would take over from the mini computers produced during the 1970s. Lechner replied that he expected to see a wide range of high-end and low-end computers on the market. He pointed out that software would play a key role in the ongoing evolution of hardware. Kildall noted that smaller systems would emphasize "real-time control" by the user rather than the time-sharing that was common with larger mainframes.

Lechner and Kildall also commiserated over the rapid changes seen in the computer industry during the preceding 30 years. Lechner said he was part of the team that developed the IBM 702 mainframe back in the 1950s. At the time, Lechner said it took a team of 17 people working for nearly a year to install a machine like the 702–and they wondered then if they could even sell 30 such systems in the entire United States.

## It's Like a Rolodex…But on a Computer!

The final segment of this first episode was also the first-ever product demonstration. Cyril Yansouni joined Cheifet, Kildall, and Lechner to show off the HP-150 personal computer. Yansouni was then general manager of HP's personal computer group. The main selling point of the HP-150 was its interactive touchscreen. As Yansouni noted, touchscreens were not entirely new technology, even in 1984. They had been used on terminals. But what HP offered was a machine that embedded the touchscreen's functionality with software applications.

The specific demo Yansouni presented was for an "electronic Rolodex" program where the user could use the touchscreen to manually scroll through virtual contact cards. He showed how the user could print out an individual card using a built-in printer. Alternatively, if a modem was attached, the user could touch a phone number to call their chosen contact.

Kildall noted that one criticism of touchscreen-based interfaces was that holding up a finger all the time can be tiring for the user. He joked that HP might want to offer an armrest attachment. Yansouni acknowledged that the "ideal" user interface had not yet been developed. He said further experimentation was necessary, and that in the future we would likely see a combination of things, including mouse-based input and voice, that could be used with a specific application.

Cheifet then pointed to yet another "new" feature of the HP-150–disk drives that supported micro-floppies, i.e., 3.5-inch diskettes as opposed to the industry standard 5.25-inch disks. Yansouni said HP's micro-floppies had the same density and storage capacity as 5.25-inch disks, and over the next several months he anticipated an improvement in capacity up to 1 MB.

The segment, and the show, ended with a brief colloquy between over the likely "next step" in computer evolution. Lechner said one trend to keep an eye on was networking computers together, both in the form of local area networks (LANs) and over telephone lines. Kildall said he expected most of the advancement in this area would involve the latter, as LANs were expensive and had a lot of problems, while telephone-based networking could rely on existing infrastructure. Yansouni agreed with Kildall, adding that telephone-based computer networking could help promote the use of voice interfaces.

## The Press Reviews of the HP-150

Hewlett-Packard clearly saw the HP-150 as their ticket to success in the still-developing personal computer market. An October 1983 cover spread in *Byte* magazine went into much greater detail about the computer–which had the internal code name "Magical"–and its touchscreen capabilities. HP actually trademarked the technology under the name HPTouch. It was actually an optical touchscreen that relied on infrared beams to detect finger placement. There were 24 holes drilled into each side of the screen's bezel, and 40 holes drilled into the top and bottom, which created a grid for the infrared beams.

The HP-150 was also an "all-in-one" PC like the Macintosh, with the computer and monitor functioning as a single integrated unit. Unlike the Macintosh, however, the disk drives had to be attached and connected separately. But as Yansouni pointed out during the on-air demo, it was possible to attach an integrated thermal printer into the base unit.

That said, another review, published in the March 1984 issue of New Zealand's *Bits & Bytes* magazine, noted that "there are no extras" included with the HP-150 and that everything, including the internal printer, had to be purchased as a separate option.

And these options added up to a very expensive machine even for the time. *Bits & Bytes* reported the base cost as NZ$8,460. The dual 3.5-inch drives that Younsani showed in the *Chronicles* demo would then run you an additional NZ$2,885 (or NZ$2,054 for just a single drive). The Personal Card software–i.e., the Rolodex program–was also sold separately for NZ$262.

To make things more difficult, the *Bits & Bytes* review noted, the HP-150 only worked with Microsoft's MS-DOS. It did not offer any alternative operating systems, notably Kildall's CP/M, which was then the standard for many business computers. The HP-150's operating system would allow users to copy data from CP/M machines, but not the actual programs. As a result, an existing CP/M user who purchased the HP-150 would also need to replace all of their existing software library.

Additionally, even though the HP-150 used MS-DOS, it was not 100 percent IBM PC-compatible. Jim Sutton, one of the lead designers on the HP-150, explained to *Byte* that software written for the IBM was "transportable" to the HP-150, so long as the program did not depend on "special hardware features of an IBM device" and only relied upon "vanilla MS-DOS calls."

Within a couple of years, HP would indeed move away from the HP-150 and its touchscreen interface to produce more mainstream IBM compatibles, notably the HP Vectra line. The original HP-150 demonstrated on *Chronicles* was also quickly supplanted by the HP-150 II later in 1984, which only kept the touchscreen as an option.

## Cyril Yansouni's Disk Drive Adventures

Cheifet and Yansouni's exchange regarding the implementation of 3.5-inch floppy disk drives on the HP-150 turned out to be an interesting bit of unintentional foreshadowing regarding Yansouni's own post-HP career. Yansouni originally joined HP in 1967 and remained there until 1986, when he moved on to a succession of corporate jobs in the tech industry, culminating in his 1991 appointment as chairman and chief executive officer of of Read-Rite Corporation, which was a top supplier of thin-film recording heads used in disk drives.

During Yansouni's first year as CEO, Read-Rite's market share rose to nearly 50 percent. But in the mid-1990s, the company stumbled when it pissed off its biggest customer, Western Digital. According to a 1997 report from the *San Francisco Chronicle*, Yansouni decided to pursue the next generation of disk-drive heads rather than expanding the capacity of its existing models. Western Digital, which favored the latter approach, quickly shifted its orders to Read-Rite's competitors. This in turn prompted Yansouni to reverse course and admit he made a mistake, but not before the company posted a $43 million loss and had to fend off a hostile takeover bid.

Read-Rite rebounded from these events and continued in business for a few more years before succumbing to bankruptcy in 2003, as described in *The Register*:

> Read-Rite ran into trouble early in 2002 when the company's heads proved not to work to well in 40GB-per-platter drives. Hard drive makers turned away from the company, and its cash quickly began to drain away. Its 2002 fiscal year saw a 63 per cent drop in revenue from which the company never recovered.

As it turned out, Western Digital acquired the remaining assets of Read-Rite for $95.4 million. After his company's demise, Yansouni went on to serve on a number of corporate boards. As far as I can ascertain, Yansouni is still alive as of this writing at the age of 78.

## Notes from the Random Access File

- This episode was recorded at KCSM-TV on October 31, 1983, and broadcast two months later on February 5, 1984.
- You can watch this episode at the Internet Archive.
- If you would like to see a full demonstration and tear-down of the HP-150, Adrian Black of the YouTube channel Adrian’s Digital Basement posted just such a video in September 2020.
- During their podcast interview, Leo Laporte mentioned he actually worked with Stewart Cheifet at Hewlett-Packard “a long time ago.”
- SRI’s Herbert Lechner will be a recurring guest (and occasional host) on *The Computer Chronicles* during this first season.
- KCSM-TV, where Cheifet produced *Chronicles* as general manager, was originally established by the College of San Mateo in the 1960s as a student broadcast training facility. In 2017, Sonoma County public television (KRCB) acquired KCSM and changed its call letters to KPJK. The KCSM call sign is still used for an FM radio station that specializes in jazz programming.
- During this first season, the presenting sponsor for *The Computer Chronicles* was Micro Focus, an enterprise software company that is still in business today.
- The Computer Museum of Boston, the subject of the first feature segment, continued in operation until 2000, when most of its collection was re-homed at the Computer History Museum in Mountain View, California.
- As you can probably tell, I intend this to be the first in a series of blogs that, ideally, will cover every episode of *The Computer Chronicles*. This might take awhile.

# Chapter 2: Visi On vs. the Apple Lisa

There was apparently a roughly two-month gap between the taping of episodes of *The Computer Chronicles* in late 1983 and their initial airing in early 1984. Looking back 37 years later, this gap may not seem that significant. But in just the second broadcast episode, it may be that *Chronicles* unintentionally provided information that was already out-of-date to its PBS audience.

## Integrated Software — The Descendants of Xerox

The subject of this episode was "integrated software," i.e., a suite of business programs that all work together. Today, most of us intuitively associate integrated software with Microsoft Office, but that product would not exist for another six years. Indeed, Stewart Cheifet opened the host segment by reminding the audience, "And when we mention integrated software, you probably think of Apple's Lisa," the business-focused computer released in January 1983.

Cheifet and Gary Kildall briefly bantered about the precise definition of integrated software. Kildall explained there were really two things involved: integrated applications and the use of a visual interface. He noted that the subjects of the episode's product demos, the Lisa and Visi On, were both descended from work done by Xerox in the 1970s, specifically the idea of an overlapping windowed graphical interface developed by legendary computer scientist Alan Kay.

Cheifet then provided narration over some especially unremarkable B-roll footage. He reiterated that integrated software relied on a graphical interface that could split up the screen and create overlapping windows to enable multi-tasking. The long-range objective of such software was making computers more manageable and easy to use. Cheifet added that such software required a “low level of artificial intelligence” and that the simper such programs were for the user, the more complex they were for the machine. And many observers still believed that the computers of present-day 1983 were “too immature” to use integrated software effectively.

## John Couch Defends “Cutesy” Lisa’s Approach to User Interface

For the next segment, former Apple executive John Couch joined Cheifet and Kildall. Couch had been vice president of the Lisa division at Apple and left the company to start his own retail store called Computer Kingdom. Kildall opened the discussion by mentioning the similarity of the Lisa to Xerox’s work from the previous decade. Couch noted he was hired by Steve Jobs at Apple back in 1978 as part of the company’s “new products” group, where the goal was to “define the personal computers of the 1980s.” Couch and his team were tasked with addressing the growth of the personal computer industry in the wake of the Apple II line’s success.

In particular, Couch said that existing machines were too difficult to use and relied on knowledge of programming languages and a command-line interface. To the extent there was integration, it was only at the file level. Couch said he had recalled seeing Xerox’s earlier work when he was in graduate school and concluded it was possible to replace the command-line interpreter with hi-resolution bitmap graphics that would “emulate how people work in offices.”

Kildall pointed out that Apple’s real breakthrough with the Lisa was in terms of price. The Xerox Star workstation, released in 1981, was prohibitively expensive for most businesses. The Apple Lisa, in contrast, was relatively less expensive. (The Star’s introductory price was about $17,000, while the Lisa initially sold for around $10,000.) Couch replied the real benefit of the Lisa over the Star was that Apple had managed to displace the command-line interface entirely. The Star also contained “very little integration” in his words, at least in term of moving data between different applications. Apple’s goal with the Lisa was to build a software architecture that would define the path for data movement and rely on a graphical interface.

Cheifet then asked Couch about what kind of user he had in mind when he developed the Lisa. In a typically Apple response, Couch said the target users were the developers themselves. Echoing the legend of Jobs and Steve Wozniak—they couldn't buy a computer they wanted so they built one themselves—Couch said they developed an integrated computer that a user could learn to use in less than a half-hour. Couch emphasized the importance of the Lisa project in penetrating the business market, pointing out there were "still 32.5 million people" working in offices who could benefit from the use of a personal computer, but they were limited by the time it took to learn a machine like the Star.

Kildall retorted that some folks were critical of the Lisa's half-hour learning time, dismissing Apple's graphical interface as relying on "cutesy" gimmicks like an animated trashcan. Couch said he didn't see a problem. The garbage can was simply an alternative way to delete files. And he rejected Kildall's implicit argument that the Lisa was not a serious business machine. Couch noted that he still used the Lisa to run his own business. To that point, Couch offered a demo of the Lisa using his company's sales spreadsheets.

The demo itself was basically Couch showing how cut-and-paste works, selecting a column of data from the sales spreadsheet and pasting it to the Lisa's graphing program to make a pie chart. Cheifet had to end the segment for time, quipping, "I know it took you three years to figure this out and I only gave you 3 minutes to demonstrate it."

## Bill Coleman Offers a Competing Visi On

In the final segment, Bill Coleman joined Cheifet, Kildall, and Couch. Coleman was the group manager at VisiCorp, which developed a rival integrated software package called Visi On. Similar to the Lisa's operating system, Visi On provided a graphical interface with integrated applications, albeit on top of MS-DOS. (Keep in mind, Microsoft Windows would not be released for another two years.) Coleman jumped right into his demo, showing how Visi On’s Applications Manager could manipulate windows, launch programs, and bring up the built-in help pages. He also mirrored Couch’s demo, copying data from a spreadsheet and pasting it into an integrated graphics application.

Cheifet then asked Couch how these types of integrated software packages differed from Lotus 1-2-3, then a popular spreadsheet program for the IBM PC. Couch explained that Lotus was a single application. It did not offer any integration with a larger accounting package or other software. In contrast, both Lisa and Visi On were “systems themselves,” where additional applications could be installed on top of what was provided.

Kildall asked if this meant that the Lisa and Visi On were “closed systems.” Coleman said no, both were actually open systems that provided toolkits for third-party developers to write their own applications. Couch agreed. He noted that while the Lisa came with seven pre-installed applications, its toolkit could be used to write additional software, such as a legal or accounting package.

Kildall asked both guests how their software compared with DESQ, a task-switching program that like Visi On worked on top of MS-DOS. Coleman said DESQ had little or no integration and “very, very little capability” to transfer data between programs. Couch added that DESQ only offered limited integration at the file level.

Finally, Cheifet asked where software integration was going. Couch replied that rather that simply transferring data through a conduit like the Lisa, he saw future systems relying more extensively on database management systems. This would allow data to always reside in a central location while the software would define pathways to cut and “can” that data to create new applications. (This process sounds an awful lot like modern APIs.)

## “The Biggest Problem for the Lisa Was Apple Itself”

As I alluded to in the introduction, the gap between the recording and initial airing of this episode would seem to have been unfortunate, largely due to the focus on the Lisa. In January 1984, a month before this episode aired, Apple released the original Macintosh. This basically relegated the Lisa to the dustbin just a year after its own release. Longtime Apple podcaster Stephen Hackett wrote on his 512 Pixels blog in 2010:

> The biggest problem for the Lisa was Apple itself. In January 1984, the Macintosh was released at a much lower $2,500 price tag. The Mac had many more features than the Lisa and a larger software library. The Macintosh outsold the Lisa very quickly, and was soon the system of choice for Apple customers.
>
> Four short months later — April 1984 — Apple released MacWorks, which allowed the Lisa to emulate the Macintosh ROM. This solution was based entirely on the floppy drives — it could not use a hard disk drive for storage. To call it rough is an understatement.
>
> One year later, Apple released the Macintosh XL. In reality, the XL was just re-badged a Lisa that ran Macintosh System 5 in emulation (via MacWorks XL) in a much smoother fashion than the Lisa machines, including hard drive support.

Hackett's reference to the Macintosh's "larger software library" may have been something of an understatement. From what I can tell, there was essentially no third-party software ever developed for the Lisa. John Couch told Gary Kildall that the Lisa came with its own development toolkit. It was actually an entirely separate operating system called the Lisa Workshop, and it ended up being used to develop early Macintosh software.

## VisiCorp's Quick Collapse

For what it's worth, Visi On did not fare much better than the Lisa. Robin Webster reported in the October 2, 1984, issue of *PC Magazine* that after suffering "legendary problems" in actually trying to bring Visi On to the market, VisiCorp "has called it quits" and sold the source code to Minneapolis-based Control Data Corporation (CDC). Webster said CDC had no plans to sell Visi On to end users—Visi On retained any retail rights

—but would instead integrate the product into its own information services network.

Webster noted that Visi On was criticized “for being too expensive and too slow,” as well as requiring an excessive amount of memory and storage for the time—512 KB of RAM and 5 MB of hard disk drive space. VisiCorp itself was also “strained” under the pressure of developing a complex integrated software package like Visi On. One unidentified source told Webster, “Our development cycle was too long; we were not put in a position to put all of the Visi On pieces together in time.”

This may have been an allusion to another key reason for Visi On’s failure to thrive in the market: the looming threat of Microsoft’s Windows. John Markoff reported in the October 1983 issue of *InfoWorld* that “[l]ess than two weeks after VisiCorp started shipping its long-awaited Visi On integrated software package, archrival Microsoft has fired an answering volley by introducing a lower-cost window manager with new features tat both users and applications-program developers will love.” Microsoft formally announced Windows on November 10, 1983, just a few weeks after Coleman taped his *Chronicles* appearance. Although Windows 1.0 did not actually ship until November 1985 — and it was not well received at that time — the writing was on the wall for VisiCorp. Microsoft’s premature announcement was enough to draw the attention of key hardware and software developers. One software developer told Markoff:

> We had to make a decision between VisiCorp and Microsoft. We decided to go with Microsoft because we think windows should be part of the operating system. Today, Microsoft is the company in the operating system business.

Even with the lengthy delay before customers actually saw Windows 1.0, VisiCorp was already done. Aside from the failure of Visi On, VisiCorp was also involved in a disastrous lawsuit against Software Arts, the company that actually developed VisiCorp’s signature product, the spreadsheet VisiCalc. VisiCorp published VisiCalc and later released other products under the “Visi” brand. Dan Bricklin, a co-developer of VisiCalc and co-founder of Software Arts, later relayed his account of what happened on his personal blog:

> As the success of VisiCalc became apparent in early 1980, Personal Software proposed a merger with Software Arts in mid-March. Around the same time, Personal Software decided to obtain venture capital funding, which it received in May of 1980, and brought in a new person to run the company. Talk of merger stopped (to the surprise of Software Arts, which was working hard to prepare for it) and relations between the two companies soured. Exactly how this change came about within Personal Software, I don't know. The talk from senior Personal Software executives, as we heard it, turned to how to lower royalty rates and then to a buy-out by Personal Software of VisiCalc outright for a lump sum. Animosity between the top executives at the time led to a change to negotiations done by second level executives. Negotiations moved slowly.

Bricklin said VisiCorp sued Software Arts in September 1983, right around the time VisiCorp was marketing Visi On, alleging that Software Arts was late in delivering an updated version of VisiCalc. Software Arts then counter-sued for breach of contract. Eventually, the lawsuit settled in the summer of 1984, with "VisiCorp paying Software Arts some money for past royalties and giving up the VisiCalc (but not"Visi" alone) trademark." This would also prove to be the end of the line for VisiCorp as an independent company, as in November 1984, it merged into Paladin Software, which dropped the "Visi" branding as it was considered damaged beyond repair.

## Couch Returns to Apple, Retires to the Vineyard

As for John Couch, I guess things didn't work out with his computer store. Sometime later in 1984 he decided to take over a Christian school in California. Couch reemerged in the tech industry during the 1990s, first serving as CEO of a biotechnology software company and then returning to Apple as the company's first vice president of education during Steve Jobs' second tenure as CEO. Couch continued to work at Apple until the late 2010s, when he retired and bought a wine vineyard.

## William T. Coleman III (1947 - 2020)

Bill Coleman is no longer with us. He passed away on November 29, 2020, from pancreatic cancer, according to the *Los Altos Town Courier*. Coleman was a former officer in the United States Air Force who earned his master's degree in computer science and computer engineering from Stanford University. After leaving the Air Force, Coleman joined VisiCorp and helped oversee the development of VisiCalc.

In 1995, Coleman co-founded and served as chairman of BEA Systems, an enterprise software company, which was sold to Oracle in 2008 for $8.5 billion. Coleman spent the final years of his career working in venture capital, including stints as CEO of the Cassatt Corporation and Veritas Technologies. Coleman and his wife also founded the Coleman Institute for Cognitive Disabilities at the University of Colorado.

## Notes from the Random Access File

- This episode was recorded at KCSM-TV on October 24, 1983, and first broadcast on February 12, 1984–three weeks after the famous "1984" ad announcing the launch of the Macintosh.
- You can watch this episode at the Internet Archive.
- If you are interested in learning more about the development of Visi On, the June 1983 issue of *Byte* Magazine published an extensive "guided tour," featuring an extended interview with Bill Coleman.
- John Couch has a forthcoming memoir about his time at Apple, *My Life at Apple: And the Steve I Knew*, scheduled for publication on July 13, 2021.
- Not to forget the other Steve, Couch's winery sells a 2019 Chardonnay named in honor of Woz.
- Seriously, Couch should sell the wine and the book as a package deal.
- Despite the Apple Lisa's short life, there was apparently an Apple Lisa Association that published its own newsletter called *ICON*. The October 1983 issue compared the relative cost of the Lisa with the IBM XT, one of the machines targeted by Visi On, and found that a "complete Lisa" with "software, hardware, and a dot matrix printer is now only $8,400 compared to the IBM at $9,229 without a printer."
- For all the emphasis tat modern Apple places on making their devices thin and light, it's amusing to note the Lisa weighed in at 48 pounds.

And that’s not even counting the keyboard, which added an additional 4 pounds.

- Ronald K. Lakis was the director of *The Computer Chronicles* during this time. Lakis’ only other IMDB credit is for a documentary, *The UFO Experience*, produced in 1982 for KPIX-TV in San Francisco.
- One thing I would love to ask Lakis about is his practice of cutting to shots of Stewart Cheifet nodding while the guests are answering a question. It’s happened like 3 or 4 times in the episodes I’ve watched so far. I may need to start keeping a running count.
- May 21, 2021, addendum – After I published this post, Stephen Hackett and Quinn Nelson released an episode of their tech history podcast *Flashback* discussing the Apple Lisa.

# Chapter 3: Music Construction Set and the Alpha Syntauri

We begin this episode of *The Computer Chronicles* from February 1984 with Stewart Cheifet plunking on an unspecified model of Casiotone keyboard. Cheifet remarked to Gary Kildall, “This is an example of computer music,” which was this week’s subject. Cheifet added that the Casiotone could play special ROM chips that contain “popular songs” in electronic form.

Cheifet asked Kildall to explain how a computer makes music. Kildall replied that while the Casiotone was not a “general purpose computer,” contemporary personal computers like those manufactured by IBM and Commodore have “tone generation capability.” Essentially, the user could write a program to produce a series of tones and add information regarding their frequency and duration. Indeed, there was now software available that was comparable to word processing programs, but for music instead of text.

Cheifet then transitioned into the episode’s pre-recorded feature, which showcased the work of MIT’s Experimental Music Studio, which was founded in 1973 by Barry Vercoe. MIT itself described the studio as “the first facility to have digital computers dedicated to full time research and composition of computer music.” The *Chronicles* segment specifically discussed the studio’s use of computer systems to “dissect” notes into their soundwave components. A composer could then adjust the notes physical characteristics “independently and instantly.”

Specially developed musical languages allowed for the creation of music in a digital form that could then be stored on magnetic tape. This went “beyond the range” of what mechanical instruments and the human voice could accomplish, Cheifet said in narration. He concluded that the next step would be to implement “real-time production synthesis,” i.e., live performances using computer instruments. This, however, would require high-speed computers capable of performing “several hundred million calculations per second.”

## Composing Music on an Apple II (with an Inexpensive Software Package)

Will Harvey and John Chowning joined Cheifet and Kildall for the next segment. Chowning, who discovered the algorithm for FM synthesis in 1967, was then the director of the Center for Computer Research in Music and Acoustics at Stanford University. Kildall opened the segment by asking Chowning about the state of computer generated music. Chowning noted that it had been 20 years since scientists first conceived of the idea of computer music, and in recent years there had been “large-scale integration” with personal computers. Moving forward, Chowning said we could expect “more and more power in smaller and smaller units.” Practically speaking, this meant computer music that was once possible only in the laboratory could soon be put into concert halls or other small environments.

Cheifet pointed out that some people were critical of the combination of computers and music, arguing the former diminished the “human creativity” required for the latter. Cheifet asked Chowning to clarify what role computers actually played in creating music. Chowning noted that computers relied on programming languages, which themselves represented “10,000 years of man’s thought about thought.” This meant that once a computer language was involved, an individual user has “access to a degree of power and generality that was not the case with ordinary musical instruments.” He pointed to his fellow guest, Will Harvey, who managed to develop a musical composition program using just a personal computer and a programming language.

Cheifet followed up by asking if that meant computers actually helped with the “composition process.” Chowning said yes, the programming language was an “enormously rich resource” that enhanced the composition process.

With that, Cheifet turned to Harvey, who developed *Music Construction Set*, a program then sold for about $40 by Electronic Arts. Harvey explained that with his program, a person who didn’t know anything about music–and may, in fact, be intimidated by learning the subject–could still create their own music using nothing more than a joystick. Cheifet then asked Harvey to demonstrate *Music Construction Set* using an Apple II. Essentially, the user moved a hand-shaped cursor to pick up notes and set them down on a musical staff. After setting up the notes, the user could then select various actions, such as playing the assembled composition. Harvey played a demonstration song that came bundled with the software. In response to a follow-up question, Harvey added that the notes scrolled by as they played. The composition screen was basically a digital representation of sheet music–only instead of using a pencil to erase notes, with his program you could simply move the notes around on the screen.

Kildall observed that software like *Music Construction Set* made learning about music more fun. Harvey noted that aside from learning how to use a joystick, which wasn’t that difficult, it was easy for users to simply play around with compositions. For example, they could make changes to works by Mozart or Bach just to see what they sounded like. Kildall asked if people actually liked to learn about music this way. Harvey said people were “overly cautions when first using a computer program.” But after about 5 or 10 minutes, they realized it was fun and they wouldn’t hurt anything.

Cheifet then chimed in, pointing out that while *Music Construction Set* was targeted at people who didn’t know much about music, you still had to take notes and place them on a staff. How could you do that when you didn’t know anything about music to begin with? Harvey replied the manual that came with the software explained basic music notation and how the staff works. And if the user was unsure about what a note sounded like, they could just press a key to hear any selected note.

Cheifet concluded the segment by asking John Chowning how software like *Music Construction Set* compared with the work he was doing at Stanford. Chowning said that it was assumed that participants in his program already understood the “abstract symbols” associated with music. Indeed, the activity at CCRMA was centered on musicians who had already studied for some considerable time and were now focused on the engineering aspects of computer music, as opposed to drafting sheet music. Nevertheless, Chowning described Harvey's software as "interesting" and said it provided a "very accessible medium" to learn the abstract language of music notation.

## Composing Music on an Apple II (with an Expensive Keyboard Attachment)

For the final segment this week, Chowning remained with Cheifet and Kildall as they welcomed Ellen Lapham, the president of Syntauri Corporation. Lapham demonstrated her company's signature product, the Alpha Syntauri, an electronic keyboard that connected directly to the Apple II. Kildall opened by asking Lapham to explain how her product differed from Will Harvey's *Music Construction Set*. Lapham quipped they were both in the music business and both used computers. But the key difference was that the Alpha Syntauri was designed to turn the personal computer into a musical instrument. It allowed the user to "play, transform sounds, compose, and even learn keyboard."

Lapham showed how the Alpha Syntauri made it possible to use the Apple II like a record player, as composition files were stored on standard floppy disks. She demonstrated a composition that she recorded a couple of weeks earlier. The bundled software then displayed the music as it played back. Unlike *Music Construction Set*, which only displayed music on the staff, the Syntauri software could also display an on-screen keyboard. Lapham noted this could also be used to learn simpler songs.

Cheifet then turned to Chowning and noted the computers he used at Stanford were "so large you couldn't bring it to the studio." But Chowning did provide an example of the output produced at Stanford in the form of a manuscripting program developed over the past 10 years by Stanford's Leland Smith, which made it possible to print musical compositions using a standard printer plotter. This software effectively allowed composers to self-publish their own composition.

Cheifet then introduced an audio recording provided by Chowning, which offered some examples of the "high-quality vocal synthesis" technology developed at Stanford. In simple terms, these were samples of computer-generated tracks that mimicked human singing. Chowning explained that it took researchers about six months to identify the "natural cues" that often seemed lacking in synthesized music. He demonstrated the multiple layers that the computer needed to introduce in order to produce something that sounded like natural song.

Cheifet then interjected to introduce another pre-recorded segment from the MIT Experimental Music Studio–a piano-computer duet of "Moments Musicaux," composed by Martin Brody and accompanied on piano by David Evans.

After the musical interlude, Kildall asked Chowning if what we'd seen in computer music is still "mostly in the lab" or if it was being used in production. Chowning said that thanks to large-scale integration, computer music had found applications in industry. For instance, there were now musical synthesis algorithms that could be integrated into circuits, such as those used in Lapham's Alpha Syntauri.

Cheifet then asked Lapham to elaborate on the use of floppy disks as a substitute for traditional music albums or cassettes. Did she see floppy disks as a new form of music publishing where people used their computers as the playback system? Lapham replied absolutely, she used floppy disks to self-publish her own compositions. The personal computer allowed her to write a song to disk and "send it to people around the world." This enabled every musician to become a publisher. Music teachers could even publish their own material.

Cheifet ended the program by asking Chowning if he saw any "resistance" from professional musicians and composers to the use of computers, or if they actually loved this new technology. Chowning answered that "virtually everyone" who came to study music at Stanford learned about computer music. This also helped to make them exceptional programmers. In fact, Chowning said there was "very little resistance" to computer music among the younger generation–just look at what Will Harvey had done.

# Will Harvey: From Boy Programming Whiz to Man Running Another SasS Company

Although it’s still early in the run of *The Computer Chronicles*, it may already be safe to say that Will Harvey will turn out to be its youngest in-studio guest. Harvey was just 15 when he originally programmed *Music Construction Set* for the Apple II, and he would’ve been around 16 or 17 when he taped this episode. Harvey actually ended up attending John Chowning’s school, Stanford, and ultimately earned a doctorate in computer science.

In a profile for the September/October 1984 issue of MicroTimes, a California-based computer magazine, Mary Eisenhart said Harvey began programming on the Commodore PET when was 12. One of Harvey’s earliest programs was a grading system that helped his mother in her work as a college instructor. Harvey switched to the Apple II platform soon thereafter, where he developed a game called *Lancaster*. Harvey told Eisenhart he “didn’t know anything about music” when he was making the game, so he “had to write a program that would allow me to convert some sheet music that I’d bought at a store to the pitches and duration that a computer could put a sound to.”

This “rudimentary transcription program” ended up becoming *Music Construction Set*. Eisenhart said that when Electronic Arts met with Harvey, the publisher was less interested in his *Lancaster* game than his music program. Harvey and EA then “spent eight months” refining the software to give it “additional features to make it attractive to the professional musician and serious hobbyist as well as to the beginner.”

*Music Construction Set* was a success on the Apple II and later ported to the Atari 8-bit computer line, the IBM Personal Computer, and the Commodore 64. Harvey told Eisenhart that he was “especially enthusiastic about the Commodore 64” because it had “the most sophisticated sound available in a home computer.” Harvey said he also planned to revise his software to make it compatible with the then-new MIDI format, which was becoming popular among professional musicians.

Harvey's post-*Music Construction Set* and post-Stanford career proved to be quite active. Video game journalist and historian Frank Cifaldi wrote in 2005 that Harvey founded the company Sandcastle in the mid-1990s, which developed networking tools for early multiplayer games. After selling Sandcastle to Adobe in 1997, Harvey "founded another company called *There*," which was an attempt to "create a virtual world for online socializing." Harvey told Cifaldi he left There in 2003 after investors pushed the company to focus "on creating simulation software for military situations as a defense contractor."

Harvey followed up There in 2004 by co-founding IMVU, a social network that is still in operation today. IMVU was in Beta when Harvey spoke to Cifaldi, who noted Harvey sounded "genuinely excited when discussing the game's economy." IMVU allows users to communicate with one another via 3-D avatars. Harvey also touted the ability for users to create and sell items, with the proceeds split with IMVU. (If this sounds familiar, it's basically the same concept as *Second Life*, which debuted around the same time as IMVU.)

Today, Harvey is the CEO of Finale Inventory, another company he co-founded in 2009. In contrast to his previous, games and social-media driven businesses, Finale Inventory is a software-as-a-service company specializing in inventory management solutions for Fortune 500 businesses.

## Ellen Lapham Continues to Climb

The other product demonstrated in this episode, the Alpha Syntauri, did not appear to have as long of a shelf life as *Music Construction Set*. I could not find any significant references to the product after 1984, and the Syntauri Corporation itself did not seem to last past the mid-1980s.

The most notable piece of press I did find was a New Hampshire-based computer magazine called *SoftSide*, which reviewed the Alpha Syntauri in its October 1982 issue, about a year before Ellen Lapham appeared on *Chronicles*. The reviewer, Steve Birchall, compared the Alpha's manual to a Phillip K. Dick novel (specifically, *Ubik*), noting that learning to use the system "requires quite a lot of homework, listening and practice, similar to learning a new word processing program.” That said, Birchall found the Alpha Syntauri “musically worthwhile” and provided features that “gave me a new freedom to jump from one sound to another instantly, which I never had with the analog equipment.” Birchall ultimately concluded the Alpha was “the way to go” for professional musicians.

The price certainly was professional. At the time of Birchall’s review, the complete Alpha Syntauri system retailed for $1,795. And keep in mind, this was not a standalone unit. It required an Apple II computer “with monitor, one disk drive, game paddles,” and an audio system to function properly. And how much would that run you? Perusing through some of the computer store ads from that 1984 issue of *MicroTimes* referenced earlier, you could get an “Apple IIe package” from one retailer for $995. That would get you the computer, one disk drive, and the monitor. Not sure what a separate audio system and game paddles cost.

As for Syntauri’s president, Ellen Lapham, she’s also a Stanford graduate (she earned an MBA). And like Will Harvey, Lapham seems to have bounced between a number of founder/CEO jobs during her tech career. According to her Facebook page, she ran Award Software International, Inc., and Searchbutton Corporation. Today she seems to have turned her attention to promoting outdoor interests, particularly mountain climbing. Lapham is co-founder and chairperson of the board at the American Climber Science Program, a Colorado-based nonprofit organization that facilitates “research and conservation in remote and mountain environments and provide opportunities for education and true exploration.” She also co-founded the Sustainable Summits Initiative, which hosts biennial conferences that focus on developing solutions to the problems caused by human impact on mountain climates.

## John Chowning’s Long Stanford Career

Finally, there is Stanford professor John Chowning. As mentioned above, Chowning invented the FM synthesis algorithm in the 1960s, essentially making him the father of the modern synthesizer. According an an online biography by Jason Ankeny, Chowning initially licensed his FM synthesis patent for one year to Yamaha. Stanford actually fired Chowning shortly thereafter, Ankeny said, due to his “meager musical output.” But after Yamaha renewed its patent license for another 10 years, Stanford “quickly rehired Chowning” and later installed him as director of the CCRMA in 1975 before awarding him a full professorship in 1979. He eventually retired from Stanford as the Osgood Hooker Professor of Fine Arts and Professor of Music.

## Notes from the Random Access File

- This episode was recorded at KCSM-TV on December 5, 1983, and first broadcast on February 19, 1984.
- You can watch this episode at the Internet Archive.
- I’m reasonably certain that the pianist David Evans featured in the MIT music segment is not the guitarist from U2 better known as the Edge.
- Ellen Lapham told her high school alumni newsletter in 2019 that she “advised” a “very young Apple Computer” back in her days as a public relations and marketing specialist.
- More impressively, Lapham also climbed Mount Everest. Twice.
- If you’re interested in Will Harvey’s first game, *Lancaster*, it’s available at the Internet Archive.

# Chapter 4: Singer Link and SOM

Normally, *The Computer Chronicles* highlighted consumer software and hardware. Stewart Cheifet often described his role as doing the legwork on behalf of the viewer so they knew what products to buy. This particular episode, however, goes in a somewhat different direction. The subject is simulator software, but aside from the opening host segment, the episode is largely devoted to proprietary software used in non-consumer applications.

## Flight Simulators – Computer Game vs. Training Tool

Cheifet and Gary Kildall opened the episode by playing Microsoft Flight Simulator 1.0 on an IBM Personal Computer. Cheifet noted this was a good example of how you could use a computer to simulate real-world situations. Foreshadowing one of the episode's guests, Cheifet added that simulation software could also be used in architecture and urban design. Cheifet asked Kildall that given everything a computer does is essentially a simulation, what do we mean by "computer simulations." Kildall replied that it's about generating scenes or a situation that someone wants to experience, which requires graphics and a lot of computing power. But it's ultimately less expensive to do than the real thing. For example, it's cheaper to run a simulation of a 747 landing than to land an actual plane.

Cheifet then narrated a short feature on the use of flight simulators by the U.S. military to train pilots in air-to-ground combat. While this might resemble a computer game, Cheifet said these simulators included "random factors" to help better simulate real-life conditions. He compared the progress of computer simulation to film animation, which managed to achieve a "remarkable level of detail" during the 1940s and 1950s. Cheifet then introduced a computer architectural model where buildings "can be viewed at almost every perspective," eliminating the need for hand-drawn blueprints.

As it turned out, both of these items were produced by the episode’s guests. The first guest was Frank Lewandowski, a senior scientist with the Link Division of the Singer Company, also known as Singer - Link. You may recognize the “Singer” name as that of a sewing machine manufacturer. It’s the same company. The Link Division developed the military flight simulation software introduced by Cheifet earlier.

Kildall opened up the discussion by noting his “second great love is aviation,” with his first obviously being computers, as that paid the bills. There was then another demo of *Microsoft Flight Simulator*, which was developed by Bruce Artwick. Cheifet noted this program was based on a Cessna. Kildall added the actual scenery in the game was rather choppy and not a “real time” picture. Lewandowski explained that *Flight Simulator* has a picture composed of 20 to 30 lines updated about twice per second. In military applications, there needs to be a “much more sophisticated scene.” A combat pilot could not accept the type of scenery seen in *Flight Simulator*. They require something closer to “8,000 edges for every TV frame time and to compute a new scene at each frame time,” rather than 2 or 3 times per second.

Kildall said that required more computing power than an IBM PC could achieve. Lewandowski agreed, noting that while a PC had a single integrated circuit, the Link Division’s simulators had 40,000 such circuits–quite a substantial difference.

Cheifet asked for more specifics about the types of flight simulations done at the Link Division. Lewandowski said that for commercial aircraft and airline pilots, the simulators were limited to takeoffs and landings. That was not that difficult to train. The more difficult applications were for the military, as those involved air-to-air and air-to-ground combat. This required simulating an entire environment, not just the mechanics of the flight. Lewandowski then narrated some footage from the Link Division’s military combat simulator, which showed a “real-time scene” of a Boeing AH-64 Apache attack helicopter firing a missile at a tank. Cheifet pointed out that the scenery was “not painted, but computer generated.” Lewandowski said yes, every part of the picture was generated 30 times per second, which was the “present state of the art.”

Kildall asked if the simulator used a database to represent the terrain. Lewandowski said it did. Everything in the simulated world, such as trees or mountains, were coded into an XYZ coordinate. Kildall noted this was what required so much computing power. Lewandowski added that over the next few years the goal was to develop "textures," so that for example you could show the "foliage" on individual trees. Or as an enemy helicopter got closer, the simulator operator could see the "other pilot" and weapons officer. And of course, you could see more textures on ground buildings and equipment. Kildall closed the segment by asking how long it would be before we would see this type of detailed scene generation. Lewandowski said some people were getting close now with the use of video disc, which mixed live imagery with computer graphics. But the Link Division's simulators relied entirely on computer graphics.

## Using Proprietary Simulation Software to Stay "One Step Ahead" of the Competition

For the final segment this week, Steve Harrison joined Cheifet, Kildall, and Lewandowski. Harrison was the data processing manager at the architecture firm of Skidmore, Owings & Merrill (SOM) in San Francisco. Cheifet asked Harrison how an architect uses computer simulation. Harrison said his firm used it for "perspective drawing," including the study of how shadows from a building affect the urban environment. Simulators could also test the effects of earthquakes on buildings, as well as the energy performance of a building over the course of a year as the sun traveled in different paths.

Cheifet asked Lewandowski how this differed from the simulators he works with at the Link Division. Lewandowski noted that architecture simulations did not have the "same constraints" as an aircraft simulator in that the former did not have to be in "real time." That is to say, a flight simulator needs to react as the pilot operates the controls, and the scene must reflect those changes immediately. Cheifet asked Harrison to clarify that something like an earthquake simulator for a building was not in real time. Harrison said no, instead they would take a tape that was made from the "actual forces" of a prior earthquake and play it back over 4 to 5 hours of computer analysis.

Harrison then showed some perspective drawings that were made using his firm's proprietary architecture simulator. The demonstration building was actually San Francisco's City Hall (which was not designed by Harrison's firm). These were wire frame drawings without much surface detail, which Harrison said were commonly used in "massing studies." Kildall asked what type of computing power was necessary to run the simulation. Harrison said it ran on a VAX 11/780 mainframe using a Tektronix 4113 color terminal. Harrison added the software could also run on the "smaller" VAX 730 and 750 machines. Cheifet asked about the software itself. Harrison replied it was a proprietary program called "Draft" developed by SOM for perspective, 2D, and 3D drawing. Harrison then showed some simulation drawings of the Louise M. Davies Symphony Hall in San Francisco, which SOM designed. Harrison pointed to the high amount of window detail in these simulator renderings.

Kildall asked if this type of simulation software was being used throughout the architecture profession or just by SOM. Harrison said this particular software was "special to his firm," but that other architects had systems "that begin to approach this." He said SOM was trying to keep "one step ahead" of its competition in this regard. There were also a number of non-proprietary systems that had some aspects of SOM's software, but his company offered an "integrated system" that applied to "all of the architectural and engineering disciplines in the firm."

Cheifet asked if SOM's software simulated the environmental impact of a structure. Harrison said it did, and in fact that was a major use of the system. He again referred to "massing studies," which Cheifet asked him to define. Harrison explained that this simply referred to the mass of a building and how it fit into a "tight urban environment" like San Francisco.

Harrison then showed a final example, this time from a shadow study of the Bank of American Building in San Francisco, then known as the Bank of America Building. Specifically, this simulation showed the shadow cast each hour by the building. Cheifet asked if it was worth the cost of a computer simulation to do such studies. In other words, Was it that difficult to study a shadow manually? Harrison replied it was, especially if you are looking at making major changes to a building. Traditionally, architects build small physical models to do these type of studies. But in a city like San Francisco where there were a lot of different structures and interaction between shadows, it was easier to test a “large number of different designs” using a computer simulation.

Kildall followed up by asking if computer simulation could also be used for interior design and the placement of objects within a room. Harrison said yes, the SOM system is actually used for interior perspective as well as exteriors. He added that interior simulations were inherently more complicated, as there were a lot more edges inside of an “open office environment,” and that was difficult to show clearly using wire frame graphics.

Cheifet turned back to Frank Lewandowski to ask what he’d like to be able to do with military simulators going forward. Lewandowski reiterated the need for “much better textures,” particularly with respect to foreground objects like rocks. Such details mattered if you wanted to train someone to drive a tank, and that was difficult to do presently with the existing “geometric textures.” Kildall asked if it was possible to “digitize” real world properties to create more realistic textures. Lewandowski said they did some of that now, such as using civil engineering drawings of airports to extract certain data about those locations, such as where to place trees, shrubs, and radar towers.

Cheifet asked Lewandowski how it was possible to simulate the human factor, i.e., the “emotional response of human actors” in a computer simulation. Lewandowski said the simulator gives the impressio of sound and motion, as well as other cues that combine with the “realistic scenery” to give the pilot everything they need. Cheifet then closed the show by asking Harrison how he would like to improve his own simulator software. Harrison joked he’d liked to get one of Lewandowski’s systems from the Link Division. Short of that, Harrison agreed that more realistic scenery was the next stage.

# The Many Links in a Long Corporate Acquisition Chain

Unfortunately, I was not able to track down anything definitive about Frank Lewandowski. I did come across a couple of obituaries that might have been for him, but nothing I could confirm. So let’s talk about Singer - Link instead. At first glance, it seems odd that a sewing machine company would have been involved in building military combat flight simulators. But during the 1960s, Singer Corporation decided to “diversify” its operations through a series of acquisitions, which included Link Aviation Devices, a company founded by Edwin Link in 1929 to market the Link Trainer (also known as the “Blue Box”), an early flight simulator he’d developed two years earlier. According a corporate history pamphlet, Edwin Link ran Link Aviation as an independent company until 1954, when it merged with General Precision Equipment Corporation (GPE). Link continued as president of the merged company until 1959. GPE then merged with Singer in 1968.

In 1981, Singer split the Link Division into two parts. The first part, Link Simulation Systems Division (later Link Tactical Simulations Division), was based in Maryland and focused on energy and industrial simulations. The second part, Link Flight Simulation Division, apparently continued to focus on flight simulators and was likely where Lewandowski worked. A few years after Lewandowski appeared on *Chronicles*, Canadian firm CAE Industries purchased Link Simulation Systems from Singer. By 1990, CAE had also acquired the Flight Simulation Division and consolidated the two divisions back into one unit. CAE renamed itself as CAE-Link.

Hughes Electronics Corporation purchased CAE-Link in 1995. Three years later, Raytheon purchased Hughes. Finally, in 2000, L-3 Communications acquired Raytheon. And despite all of the corprorate turnover, the Link simulation business continues in 2021 as L3Harris.

## Harrison Moved from Architecture to Academia

SOM, Steve Harrison’s architecture firm, is also still in business today. Skidmore, Owings & Merill dates its own history back to 1936, when it was founded in Chicago by architects Louis Skidmore and Nathalien Owings. John O. Merrill, an engineer, joined the firm three years later. Today, SOM is known for its work on a number of famous buildings, including New

York City’s One World Trade Center and the Burj Khalifa in Dubai, which is currently the world’s tallest building.

Harrison himself is no longer with SOM. He left the firm not too long after his *Chronicles* appearance. In 1985, he joined Xerox PARC as a member of its research staff. He stayed there until 2002, when he left to found two startup companies before entering academia. Since 2003, Harrison has been an associate professor at Virginia Tech’s Department of Computer Science and School of Visual Arts. He also serves as director of Virginia Tech’s Human-Centered Design Program and co-director of the Social Informatics Area for the Graduate School.

## Notes from the Random Access File

- This episode was first broadcast on February 26, 1984.
- You can watch this episode at the Internet Archive.
- *Microsoft Flight Simulator* is, of course, also still around today. Bruce Artwick released the original *Flight Simulator* for the Apple II in 1977. Microsoft later hired Artwick to write an IBM PC version of the program, which was released in November 1982. Microsoft published its most recent iteration of *Flight Simulator* in 2020 under its Xbox Game Studios division.
- SOM’s hardware setup was not cheap. The VAX 11/780 alone cost between $120,000 and $160,000, and that did not include the Tektronix color terminal.
- The Bank of America Building, located at 555 California Street in San Francisco, is no longer Bank of America headquarters. Continuing with our corporate merger sub-theme this week, Bank of America abandoned San Francisco for Charlotte, North Carolina, after merging with NationsBank in 1998. Today, the 555 California Street building is mostly owned by Vornado Realty Trust. (A 30 percent stake is owned by the 45th president of the United States.) One of the building’s current tenants is Microsoft.
- Gary Kildall said that flying was his second great passion in life. Indeed, Stewart Cheifet often mentioned in interviews that Kildall typically fly his private plane to San Mateo for *Chronicles* tapings on Saturday afternoons. Kildall’s flying passion would also factor into

one of the biggest legends surrounding the fate of his company’s CP/M operating system, which we’ll get to in a future post.

# Chapter 5: Concurrent CP/M, MS-DOS & UNIX

The first season of *The Computer Chronicles* was repackaged and marketed to educational institutions as *The Computer Chronicles Telecourse*. The recording we have for this next episode, and a few more going forward, were part of that telecourse and thus included a series of interstitial segments hosted by SRI International's Herbert Lechner, whom we met in the first broadcast episode. Lechner's segments mostly review the key concepts discussed in the regular episode and refer to an accompanying textbook for students to follow. I won't be discussing Lechner's segments during my episode recap, as it would be little more than summarizing his summaries, which would be redundantly redundant.

## Operating Systems Are "Most Exciting When They Don't Work"

Lechner also appeared this week as Stewart Cheifet's co-host. Cheifet opened the program by explaining Lechner was "sort of sitting in for Gary Kildall." In fact, Kildall was one of this week's guests, as the subject was operating systems, and he created CP/M, which Cheifet described as "perhaps the major operating system in use today."

Cheifet noted that operating systems were a tough subject to discuss. Every user dealt with an operating system, but most do not understand them. Lechner pointed out that operating systems are "most exciting when they don't work," and that's obviously not something you want to see as a user. Lechner then provided a brief overview of operating systems, explaining they manage the computer's resources and save the individual programmer "a lot of nitty-gritty effort" by providing platforms to make applications that can run on a number of machines with different configurations. And when it comes to larger machines, the operating systems have become "very sophisticated" and can manage resources in such a way as to help the computer achieve its full potential.

The episode then veered into a very unusual segment–at least by *Chronicles* standards up this point–with a business-casual Stewart Cheifet appearing to walk inside pf the enlarged interior of an Apple IIe. Cheifet greeted another man, Apple engineer Bruce Tognazzini, who explained that he's just "checking things out" with the computer before the user turns it on. (In a testament to how silly this sketch was, Tognazzini then looks at his wrist to check his non-existent watch.) Cheifet asked Tognazzini to explain what they appear to be standing in. Tognazzini explained it was the parts that make up an Apple IIe: The microprocessor, which is the heart of the computer; read-only memory (ROM) chips that contain the operating system kernel and the BASIC programming language; and eight random access memory (RAM) chips that hold information retrieved from the disk drive.

Cheifet then asked Tognazzini to explain in greater detail what the operating system does. Tognazzini replied the operating system starts to run when the user turns the computer on. (Tognazzini said the Apple IIe only used 12 volts of power when active, so he and Cheifet were in little danger of electrocution while "standing inside" the unit.) After the power is turned on, Tognazzini said the processor checks itself out before turning things over to the ROM, which has a small program that looks for a disk controller. The disk controller, in turn, has its own ROM that takes control and brings in the operating system from the disk inserted in the drive through a white cable. The operating system then takes up about one-fifth to one-fourth of the available RAM in the Apple IIe, including the BASIC programming language. After about 15 to 30 seconds, the screen displays a message informing the user they can get to work. The segment itself then ended with a pullout to the Apple IIe monitor, which displayed, "DOS ERROR – WARNING! – TWO MEN INSIDE."

## Would CP/M Remain "Simple, Clean, and Neat"?

In the next segment, Gary Kildall and Tony Fanning, a member of the corporate engineering staff at Hewlett-Packard, joined Cheifet and Lechner.

Lechner asked Kildall, the CEO of Digital Research, to explain how he got started making operating systems. Kildall said that even though he was now known as an “OS guy,” he actually started out as someone primarily interested in programming languages. During the early days of microprocessors in the 1970s, Kildall worked with Intel on a programming language for microcomputers called PL/M. It then became necessary to develop some sort of operating system to support that language, which led to the creation of CP/M, or *control program for microcomputers*. He noted that prior to CP/M, programming in PL/M required the use of paper tapes. CP/M allowed for the use of floppy disks in smaller computers.

Cheifet reiterated his point from the introduction that operating systems remained a “mystery to many computer users,” who may take advantage of its features but never delve into the inner workings. Kildall offered his own take on what an operating system does, noting there were basically three levels of software in a personal computer: Machine instructions, system software, and application software. The latter was what users actually went out and purchased in the store. Operating systems provided the system software, or the “middle layer” between the applications and the machine instruction. The OS thus principally served as a “traffic cop” that kept track of tasks like storing files and handling input and output.

Cheifet then turned to Fanning and asked what he looked for in an operating system as a user. Fanning said he never wanted to “see” the operating system. He just wanted to use applications. What you look for in an OS therefore depended on what you were trying to do. If his goal was to use a word processing program, for example, then he just wanted the operating system to “disappear” and not bother him.

Cheifet noted that Fanning worked primarily with MS-DOS. He asked Fanning to explain how that operating system differed from CP/M. Fanning said in a lot of ways, they both did the same types of things: MS-DOS also took care of files, managed data transfers, and provided a “human interface” for the computer.

Returning to Kildall, Cheifet pointed out there had now been “several generations of CP/M” released. So what was added with each new generation–i.e., how did you actually “improve” an operating system?

Kildall replied the real key was to match what you were doing with the capability of the computer system available. He noted that in the early days of microcomputers, the storage capacity was limited to 16 kilobytes (KB) of RAM. With such limited space, you needed to tailor an operating system to only use a small portion of that memory. But when you start to add more memory and faster processors, then you can add more functionality to the operating system. For instance, modern 8-bit systems now have a maximum possible RAM of 64 KB. And 16-bit computers come standard with 128 KB–moving towards 512 KB. So again, the operating systems will expand and add functions that take advantage of these larger memory sizes.

Lechner said one of the early attributes of CP/M was that it was “simple, clean, and neat.” Now that computers were much larger, would we also see the complexity grow? Kildall said it was “very easy to add bells and whistles” to an operating system that do not provide any additional functionality. The question was therefore how to add the “right kinds” of functions. He said that on the newer 16-bit machines a key feature was concurrency, also known as multi-tasking. Early microcomputers were only capable of performing one task at a time. But now a 16-bit machine could do something like print a document while continuing to run a word processing program.

Cheifet then asked Kildall to demonstrate his Concurrent CP/M operating system on an IBM PC. Kildall explained that Concurrent CP/M was a “real-time system” that could switch between programs at short time intervals. This allowed for up to four virtual consoles, each of which could run a separate program. In a business environment, Kildall said this would enable the user to run a word processor on one screen and then switch to a second screen to run spell-checking software or receive data over a modem. The user effectively had “multiple systems on your desk.” Kildall said concurrency was where operating systems were moving towards given that newer microcomputers had at least 512 KB of RAM–an amount of memory once available only on mainframes.

## A UNIX Future Led by AT&T Personal Computers?

Jean Yates joined the group for the final segment. Yates was president of Yates Ventures, a consulting group that focused on the UNIX operating system. Cheifet asked Yates to explain what UNIX was and how it compared to CP/M and MS-DOS. Yates said UNIX was another “standard operating system” that worked with a wide variety of applications and computer brands. Indeed, UNIX was unique in that it could run on anything from the IBM PC to a mainframe like the Cray supercomputer. UNIX also made it easy to connect and network different types of computer together. For example, if you had two machines that both used UNIX, the users could send email, transfer files, and even share applications between them. UNIX was also a multi-user operating system, so more than one user could access several different programs stored on the same machine using individual terminals.

Herb Lechner pointed out that UNIX originated in Bell Labs and was previously licensed only to universities and nonprofit organizations. Was UNIX now fully commercially supported? Yates said that following the breakup of the Bell System in 1982, AT&T was now permitted to sell and support the newest release of UNIX, System V, as a commercial product. That said, UNIX still needed to be “enhanced and improved” for end users, but Yates believed AT&T was making “big strides” towards that goal.

Cheifet then asked Yates to explain the difference between UNIX and XENIX. Yates said that AT&T sold two different forms of UNIX. There was the previously mentioned System V product sold directly by AT&T. Then there was the licensed UNIX source code itself, which was more expensive but allowed the purchaser to make changes to the software. Consequently, the source code was typically licensed to computer manufacturers and software companies. In the case of XENIX, Microsoft purchased the UNIX source code and “enhanced” it to run on microcomputers like the Apple Lisa, the Tandy 16/6000, and the Fortune 32:16. Yates said it was also rumored that XENIX would be sold for the IBM Personal Computer XT.

Lechner said he always associated UNIX with the PDP-11 series of minicomputers sold by Digital Equipment Corporation. He then asked Tony Fanning about his own experience with UNIX. Fanning said he’d been working with UNIX for about five years, which made him a “relative newcomer” to the operating system, although he had yet to try it on minicomputers. Fanning reiterated that presently worked mostly with MS-DOS, which he said was reputed to be moving more towards acting like UNIX. Yates added that most of the energies towards developing UNIX for microcomputers was coming from companies besides AT&T, as it again was prohibited from commercially exploiting the operating system prior to the recent deregulation. She theorized that AT&T would soon introduce its own line of UNIX-based mini- and microcomputers that offered commercial features. Right now, she said that there was still a substantial cost involved in “holding the hand” of a beginning CP/M or MS-DOS user versus UNIX.

Cheifet asked Kildall where CP/M would fit into this evolving picture. Kildall said that UNIX had been a good influence of the computer industry, especially with respect to tools and utilities. He said Digital Research was also doing work with UNIX in conjunction with Intel. Ultimately, Kildall said CP/M’s multi-tasking capabilities would prove “complementary” to what UNIX offered.

Lechner pointed out that when someone bought CP/M, they expected the operating system to remain invisible. In contrast, no two UNIX systems were alike, as each user could modify, develop, and tinker with the system software as they wished. Lechner asked Yates if that was what helped distinguish UNIX in the market. Yates replied that while CP/M was exclusively a microcomputer operating system, UNIX was derived from minicomputers and came with approximately 400 built-in programs for things like typesetting, email, text processing, unit conversion tools, and programming languages. It was by design a very big system–but also very modular. Yates said she herself deleted about two-thirds of the standard included UNIX programs because she never used them. In that sense, UNIX was a “do it yourself” operating system that required somewhat more expertise than CP/M.

## Where Are They Now?

Gary Kildall and Herbert Lechner are *Chronicles* regulars, so I won't delve into their post-show careers at this time. As for Tony Fanning, I could not find anything about him aside from a sparse LinkedIn page noting he was retired from Hewlett-Packard. So let's focus this section on Apple's Bruce Tognazzini and Jean Yates.

## Tognazzini: Old Apple Designer Becomes New Apple Critic

Bruce Tognazzini joined Apple in 1978 as one of its first 100 employees. He primarily worked on user interface design during his 14 years with the company. Tognazzini actually co-authored the demo software that introduced users to the Apple II, *Apple Presents…Apple*. More notably, he authored Apple's *Human Interface Guidelines* during the Apple II and early Macintosh eras.

After leaving Apple in 1992, Tognazzini joined Sun Microsystems and developed dozens of inventions related to human-computer interaction, including the patent for the "rolling blackout" password system commonly used on today's mobile devices. Four years later, Tognazzini joined WedMD during its startup phase as lead designer.

Today, Tognazzini is a principal with the Nielsen Norman Group, a user interface research and consulting firm. In recent years, Tognazzini has become a notable critic of Apple's user interface design choices, especially in the post-Steve Jobs era. For example, in a November 2015 article for *Fast Company*, Tognazzini and co-author Don Norman argued that Apple's user interfaces for the iPhone and iPad were "destroying design":

> Worse, it is revitalizing the old belief that design is only about making things look pretty. No, not so! Design is a way of thinking, of determining people's true, underlying needs, and then delivering products and services that help them. Design combines an understanding of people, technology, society, and business. The production of beautiful objects is only one small component of modern design: Designers today work on such problems as the design of cities, of transportation systems, of health care. Apple is reinforcing the old, discredited idea that the designer's sole job is to make things beautiful,

even at the expense of providing the right functions, aiding understandability, and ensuring ease of use.

### Yates Pivots from UNIX to Online Wine Marketing

Jean Yates originally founded her eponymous Yates Ventures in 1978 as a market research firm supporting “open systems software.” But she best became known for her work with UNIX. After selling Yates Ventures to International Data Corporation in 1981, she stayed on as a vice president with that company and ran its UNIX division based in Palo Alto, California. She also co-authored several editions of the User Guide to the UNIX System.

A couple of years after her *Chronicles* appearance, Yates retired from the computer industry and decided to open a wine store in Oregon. She would own Avalon Wine in Corvallis, Oregon, for more than 25 years. In a 2013 profile written by Andy Perdue for *Great Northwest Wine*, Yates said she managed to put her tech experience to good use in pivoting her business to online marketing during the 1990s. By 2013, Yates sold Avalon to two of her employees and started Oregon Wine Marketing, a digital marketing firm focused on small wineries.

## Notes from the Random Access File

- This episode was first broadcast on February 19, 1984.
- You can watch this episode at the Internet Archive.
- Although Stewart Cheifet described CP/M as “the major operating system in use today,” the market was already shifting away from Digital Research’s product. IBM’s adoption of MS-DOS–essentially a CP/M clone that was purchased and re-licensed by Microsoft–for its Personal Computer eventually led to it becoming the industry standard as other manufacturers quickly cloned IBM’s largely off-the-shelf hardware.
- Jean Yates’ prediction that AT&T would eventually enter the personal computer market with UNIX-based machines did come to pass. AT&T introduced the UNIX-based PC 7300 (also known as the AT&T UNIX PC) in March 1985, about a year after this *Chronicles* episode aired.

Yates was quoted in the *New York Times* stating the AT&T machine had better specs than IBM's own PC/AT.

- UNIX's legacy as a proprietary operating system would eventually lead to the so-called UNIX wars, which ended with the Berkeley Software Distribution (BSD) largely displacing System V as an open source operating system that is still widely used today.
- As for XENIX, Microsoft effectively abandoned the project once AT&T started selling System V. Microsoft eventually sold XENIX to the Santa Cruz Operation, Inc., (SCO), which continued to market UNIX-based operating systems until 2001.
- Although Stewart Cheifet and Bruce Tognazzini used an Apple IIe in their operating system sketch, there was no mention made of the actual operating system. By this time, the Apple II line had switched from Apple DOS to ProDOS. Both were proprietary operating systems used exclusively by Apple. It was also possible to run CP/M on an Apple II using a third-party expansion card.
- According to Tognazzini's website, he "created key clips" for the opening sequence of *The Computer Chronicles*. I'll discuss that opening sequence in a future post.

# Chapter 6: Wordvision, Word Plus, and the Writer’s Workbench

Since the 1990s, word processing has largely been synonymous with *Microsoft Word*. Of course, *Word* didn’t start out on top. It was first released in October 1983. At that time, the dominant word processing program was *WordStar*, which had already been on the market for several years. The first *Chronicles* episode to discuss word processors, taped in December 1983, largely framed the discussion in terms of discussing the competitors to *WordStar* yet made no mention of Microsoft’s new offering.

## Would Computer Word Processing Make Language Too Bland and Dry?

Stewart Cheifet and Gary Kildall opened this episode by looking at a large pile of word processor instruction manuals. Cheifet noted there were about 150 word processing programs currently on the market. Cheifet also displayed an example of a handheld text editing device, the TRS-80 Model 100 from Radio Shack. Cheifet asked Kildall how this differed from a word processor. Kildall explained that a text editor was something used to prepare programs. When a user wrote a program for the first time they needed the ability to go back and make changes. But a simple text editor was not good for working with finished documents. That is, they could not be used to cut-and-paste text or prepare a document for printing. That was what a word processor was for. Kildall said word processors came in a “variety of shapes and styles,” which made it difficult for many users to choose a product.

Cheifet then introduced our B-roll feature for the week, which focused on how word processors had changed the world of business writing, specifically newspapers. Cheifet noted that word processors had eliminated many of the “intermediary steps” between a reporter’s notebook and the camera-ready copy. In many instances, a story would not touch paper until the final stages of printing. Thanks to word processing and modems, an article could be transferred over phone lines from a writer's terminal to the newspaper's editorial department 30 miles away. Indeed, Cheifet said entire books were now written this way, with files stored on disk until a final draft was "shipped" via modem to the publisher.

But word processors offered more than electronic transmission, Cheifet said. They also allowed for "different degrees of interaction with the creative process," from correcting spelling errors to offering stylistic advice. As a result, there was disagreement among writers as to the overall effect that word processors would have on the quality of written language. Cheifet said some writers were concerned that computer assistance would promote "dry, bland writing" that diluted individual style. However, other writers believed that word processors would improve their ability by providing as much or as little assistance as needed.

## Did Price Matter When It Came to Word Processors?

Paul Schindler and Jim Edlin joined Cheifet and Kildall for the next segment. Schindler was a journalist then working for *Information System News*. Edlin was the designer of *Wordvision*, a recently released word processing program designed to work with the IBM Personal Computer and certain compatibles. Kildall opened by asking Schindler to describe what he looked for in a word processing program. Schindler said the most important thing for him was that the program was "easy to use" and "easy to learn," which he admitted were "popular buzzwords" used to describe many word processors. Essentially, Schindler said he wanted to start with a blank screen and write a document with minimal fuss. And since most writers, including him, were not perfect, the program also had to make it easy to correct errors and move text around, as well as format the final document in a "precise and pleasing way."

Kildall suggested the user may think of a word processor like a car where you "go to the the dealer and try it out." Schindler quipped you needed to "try several" before committing to one.

Cheifet pointed out that Edlin's *Wordvision* was priced in the $50-$70 range, which was considerably lower than *WordStar*, which could retail for as much as $500. Cheifet wanted to know how much did word processors differed to cover that price range. Schindler said they tended to differ in function. He noted that the price usually reflected performance. For example, there were some things *WordStar* could do that *Wordvision* could not (and vice versa). It basically came down to what you needed as the user. As a reporter, Schindler said he just needed to be able to "slap words on the screen." He actually used a word processing program developed by a friend. This program had no ability to format documents on the backend. But Schindler did not require such features, as he never printed his stories on paper. He sent them to a computer in New York. That said, if you required formatting and printing capabilities, you needed to find a word processing package with "sophistication" in those areas.

Kildall observed that as more companies entered the mass market of word processing software, prices would come down across the board. Edlin chimed in and said there was also a time factor. For instance, if you bought a $5 calculator from Safeway today, it would likely out-perform a $500 calculator that you purchased a decade ago. Similarly, with IBM putting out 600,000 personal computers–and counting–in a single year, the numbers at which a small software company like his could "amortize its development costs" were coming down. That meant *Wordvision* could offer more and still charge less.

Kildall said it was therefore important to look at a software package's functionality and not just its price. Schindler added the key was to not buy more than you needed. The typical writer did not require "some giant 32-bit super microcomputer" just to do simple word processing. Schindler said he'd seen articles comparing all available word processors "along an axis of 42 features." But if you only needed six of those features, buy the least expensive package that contained those six.

Cheifet then asked Edlin to demonstrate *Wordvision* on an IBM PC. Edlin clarified that he considered his software a "writing tool" rather than a word processor. By this, he meant *Wordvision* was something a person would use to work with "his or her own words," as opposed to someone like a secretary who was primarily taking down someone else's words. Edlin also said *Wordvision* was designed for "people literacy," i.e., it was about making more literate computers that understood people, not the other way around.

Edlin said this philosophy informed the user interface. He pointed to the *Wordvision* title screen, which displayed menu options using red, yellow, and green symbols to mimic a traffic light–a concept that most users would instinctively understand. Edlin also said *Wordvision* came with labels (stickers, really) that could be applied to the standard IBM keyboard to guide users on the proper commands. (He said more recent versions came with slip-over keycaps.) Files were also displayed visually so that the user did not need to type in a file name. Edlin then demonstrated the basic process of working with a document.

## Do You Want to Buy Spellcheck as a Service?

For the final segment, Warren Kuhl of Western Electric and Wayne Holder joined Cheiefet and Kildall. Holder was the principal at Oasis Systems, which developed a spellchecking package for word processors called *The Word Plus*. Kildall noted that up to this point, most word processing packages turned the personal computer into a "fancy typewriter," but Holder's software went beyond that. Holder said yes, *The Word Plus* was meant to take over where the writing process left off. When someone was engaged in the creative process of writing, they did not want to stop and check their spelling or look for redundant phrasing. His software took already created files and ran them through a series of spelling and stylistic checks before giving recommendations on how to improve the text.

Holder then demonstrated *The Word Plus* on a Kaypro 10 computer. (This machine was quite noisy and could be heard whirring throughout the demonstration.) He showed that it took about 15 seconds for the software to scan a text file and flag any spelling problems. The user could then look at each spelling error and consider alternatives from the built-in dictionary. The next step was to perform a "style analysis." Holder noted that when you're writing with a style book, there was often a tendency to fall into certain traps with respect to phrasing. *The Word Plus* could identify such traps and once again present alternatives. For instance, instead of using "SMALL IN SIZE," the software suggested just saying "SMALL."

Kildall asked for clarification as to how the program checked for possible alternatives–that is, did it actually rely on a dictionary? Holder said yes, there was an actual dictionary. Along the same lines, the style analysis used a list of phrases compiled from phrasebooks and other sources. In fact, a large part of this work was based on what AT&T had done in developing its *Writer's Workbench*, the product Kuhl was there to demonstrate. Cheifet followed up by asking if *The Word Plus* could be purchased on floppy disk to run on a microcomputer. Holder said yes, it was designed to be a "practical package" that users could purchase and use on most available word processors and microcomputers.

Cheifet then turned to Kuhl and asked him to demonstrate *Writer's Workbench*. This demo was actually performed on a terminal tied to an offsite machine running UNIX System V. Essentially, Kuhl said *Workbench* was a set of tools designed to work on UNIX to help writers. Like *The Word Plus*, the *Workbench* included both proofreading and prose analysis. He pointed out that beyond looking for things like spelling or punctuation errors, *Workbench* could also identify split infinitives. It was also designed to identify sexist language in a document and suggest gender-neutral alternatives–e.g., use "PERSONNEL" instead of "MANPOWER." The other notable feature of *Workbench* was its ability to analyze the grade-level comprehension of the entire document by looking at the mix of simple, compound, and complex sentences. For example, the sample document was assessed at a 13th-grade level of education. If this document were meant to be an "instructional text," Kuhl noted, it should be rewritten to require no more than 10th-grade comprehension level.

Cheifet asked Kuhl where the *Workbench* was currently being used. Kuhl said it was still being tested at Bell Labs and Western Electric. But it had been deployed at Colorado State University, where it had been used by about 2,500 students as part of their composition courses. Kildall asked if people would eventually become dependent on *Workbench*, to the point where if it passed its analysis, the writing would be considered good. Kuhl said that was the idea. He also emphasized this type of analysis would likely be used more in fields that required writers to follow a particular style, like journalism, as opposed to creative writing.

## Workbench's Legacy Lives on in Today's Linux and BSD Tooling

David Silverman offered some additional background on the history of *The Writer's Workbench* and its use at Colorado State as part of *Unix: An Oral History*. Silverman said the project that became *Workbench* actually grew out of an effort by two men, Bob Morris and Lee McMahon, to determine the authorship of individual contributions to the *Federalist Papers* based on text processing and statistical analysis. When *Workbench* was later "beta tested" at Colorado State, it "succeeded for three main reasons – its reliability, structure, and the programmers' understanding of the writing process." In particular, *Workbench* proved more capable than a rival offering from IBM called *Epistle*, which was "slow and incapable of coping with incorrect student grammar." *Workbench* was not a grammar checker either, but it could "illuminate the style of sentences employed," according to Silverman.

*Workbench* actually continues in use to this day, at least in spirit. During the 1990s, many of the individual tools that made up *Workbench* were rewritten for the open source GNU project. You can still download and install these tools on modern-day Linux and BSD distributions.

Wayne Holder's *The Word Plus* was similar to *Workbench* in that it offered a series of smaller routines designed to be used with another word processing program, such as the Kaypro 10's *Perfect Writer*. A 1982 manual for *The Word Plus*, also written by Holder, detailed a number of these utilities, including one to insert "soft hyphens" into words and another that calculated a document's overall word count.

And while I could not find anything about Western Electric's Warren Kuhl, Wayne Holder has had a very active post-spellchecker career. Even while his Oasis Systems continued to market *Word Plus*, Holder founded another software company, FTL Games, in 1982, as part of his Software Heaven, Inc. Holder hired an old high school classmate, Bruce F. Webster, as his lead designer and programmer. The two collaborated on FTL's first game, *Sundog: Frozen Legacy*, originally released for the Apple IIe in 1984. After Webster left the company, Holder hired another programmer, Doug Bell, who developed FTL's most famous game, *Dungeon Master*, and its sequel, *Dungeon Master II: Skullkeep*. According to *The CRPG Addict*, the first *Dungeon Master* was a "seminal game," as it was the first "real-time 3D game" released for personal computers.

Holder ran FTL until it shut down in the mid-1990s.

## Wordvision Publisher's Financial Woes Lead to Quick Demise

Perhaps the most interesting story to come out of this *Chronicles* episode was the short rise–and prompt fall–of *Wordvision*. Jim Edlin and his business partner, Bruce McGloghlin, started the eponymous Bruce and James Program Publishers in Dublin, Ohio, to self-publish their word processing program. As Stuart Cheifet noted, *Wordvision* was priced significantly below industry-leader *WordStar*, which retailed for as little as $299 at the time according to one computer magazine I reviewed from that time period. In contrast, the list price of *Wordvision* was $79.95.

But Bruce and James actually sold *Wordvision* for an even lower price originally. Before the official launch, users could sign up to buy a "Pioneer Edition" of *Wordvision* for just $49.95. This was, for all intents and purposes, an untested beta version. Stephen Manes, in his review of *Wordvision* for the April 3, 1984, issue of *PC Magazine*, quipped that Bruce and James "managed to get users to pay for the privilege of testing a product," and the included product documentation contained an explicit warning that "chances are the program will go 'flooey' and require turning the computer off and back on."

Manes was also not impressed with the final release. He said that although *Wordvision* "has more than a couple of nifty things going for it," the software was ultimately "so quirky and idiosyncratic that it may be your one-way ticket to involuntary servitude." Manes was especially critical of the "mess of stickers" that users were expected to place on their keyboard, as mentioned by Edlin during the *Chronicles* demo. Altogether, there were

28 keys that required stickers, and it was impossible to operate the software without them, as the manual constantly referred to “keys that you will see nowhere except on your Wordvision keyboard.”

Additionally, Manes noted *Wordvision* did not play nicely with MS-DOS. The user could not backup their files using the standard `COPY` command in DOS. Instead, it was necessary to load the file into *Wordvision* and then save a separate copy. As a result of its “nonconformity” with DOS specifications, Manes said *Wordvision* would not work with the just-released IBM PCjr and would likely be incompatible with the forthcoming Microsoft Windows.

Bad reviews notwithstanding, *Wordvision* was likely doomed by the time Edlin’s *Chronicles* appearance aired in March 1984. Just three months later, David Needle reported in InfoWorld that Bruce and James “has run into serious cash-flow problems.” The two co-founders were the only employees left, and Edlin told Needle that he was contemplating seeking financial support from their own customers to stay afloat. Edlin added that while *Wordvision* was selling well, the lack of money hampered their efforts to continue marketing the program. As a final indignity, Needle said Bruce and James could only afford to man their customer support line for one hour each day.

Bruce and James did manage to reorganize as a California company in May 1985, but as best I can tell, it never produced another product after *Wordvision*. A registration card included with the original release mentioned a number of potential *WordVision* add-on products, including a spellchecker and the ability to import files from other word processing programs, but none of these features were actually released.

Jim Edlin himself rebounded from the demise of *Wordvision* and had a prolific career in the tech industry. After Bruce and James closed up shop, Edlin served co-founder and co-CEO of The HyperMedia Group, Inc., which produced customer interactive media during the heyday of CD-ROMs. After selling HyperMedia in 1998, Edlin then spent a decade at Tides, a network of nonprofit organizations based in San Francisco. During the 2010s, Edlin worked with a number of tech startups and spent seven years as a senior software engineer with microlender Kiva. According to

Edlin’s personal website, he semi-retired in 2018 and now runs a small online jewelry store.

## Notes from the Random Access File

- This episode was recorded at KCSM-TV on December 5, 1983, and first broadcast on March 19, 1984.
- You can watch this episode at the Internet Archive.
- The TRS-80 Model 100 briefly demonstrated by Stewart Cheifet during the opening host segment was actually a nifty little device. According to a retrospective written by Harry McCracken for *Fast Company* in 2019, the Model 100 contained code personally written by Bill Gates and even came with a 300-baud modem. If you’d like to see a more extensive demo of the Model 100, I recommend this October 2020 video from 8-Bit Show and Tell.
- A programmer named Ethan Dicks apparently worked on a planned Commodore 64 port of *Wordvision*. In a November 2000 newsgroup post, Dicks said his first job was with Bruce and James in 1982. He prepared a short demo that was shown at the COMDEX trade show. But the actual software was “never started.”
- Stephen Manes criticized *Wordvision* because it would not allow the user to correct text by typing directly over it. This is funny, since that’s basically how all modern word processors behave–the default is to insert text rather than overwrite. I guess this was not the standard in 1984.
- On the other hand, Manes was right to criticize *Wordvision* for making it far too easy to lose an entire document with a single keystroke. This happened to me when I tested the software in DOSBox.
- According to the *Wordvision User’s Guide*, there were about 900 “Pioneers” who paid for the privilege of running the beta version.
- Paul Schindler would become a fixture of *The Computer Chronicles* for the remainder of its run, providing software reviews and even sitting in as co-host on a number of episodes.
- I kept wanting to refer to Bruce and James as Bartles & Jaymes. Sure, it’d be weird if a beverage company made a word processing program. Then again, I bet Jean Yates would approve.

# Chapter 7: Donn B. Parker and the Digi-Link

Roger Ebert wrote in his four-star review of the 1983 film *WarGames*, “Computers only do what they are programmed to do, and they will follow their programs to illogical conclusions.” In the movie, Matthew Broderick played a teenage hacker who managed to remotely access the United States missile defense system and initiate a “Global Thermonuclear War” scenario that he mistakes for a computer game. Ultimately, Ebert said the film’s message was, “Sooner or later, one of these self-satisfied, sublimely confident thinking machines is going to blow us all off the face of the planet.”

## Worrying About a “New Kind of Spy”

Several months after *WarGames* premiered to strong reviews and box office sales, *The Computer Chronicles* broadcast its first episode on the subject of computer security. Stewart Cheifet directly referenced the film in his introduction, noting that typically, computer security did not involve “something spectacular” like nuclear war, but rather more mundane things like stealing money and information.

Cheifet asked Gary Kildall if data stored on a computer was really more secure than data stored by physical means, such as inside a locked filing cabinet drawer. Kildall replied there were techniques that could make data more secure, provided they were actually used. Unfortunately, it often took a “clever individual” breaking the system before such techniques were applied. Kildall said we’re raising a “whole new generation of computer-wise people through the personal computer revolution,” and some of those people would use that knowledge in a malicious way.

Indeed, Kildall said that today–in late 1983–computer crime was already a problem. Timesharing services were particularly vulnerable because they were designed to be used by a number of strangers who made a phone call. This led into a prerecorded demonstration by Cheifet of how someone could use a computer–in this case, what appeared to be a TRS-80 Data Terminal–to remotely access a computer database system. Cheifet noted that all someone needed was a personal computer, some terminal software, and a modem to connect their telephone to the computer. Obviously, you also needed to know the phone number of the computer you wanted to access.

Cheifet demonstrated what he meant by dialing into a database from his terminal. (He noted it was a local call–not even a toll call!) He explained that you had to wait to hear a high-pitch tone, which indicated the host computer had answered. The remote computer then asked for an ID number and password. Cheifet pointed out the terminal did not display his password as he typed it. This was a security feature meant to prevent someone from stealing his password.

Cheifet also noted he was a "qualified user" and thus had proper credentials to access the remote database. But the problem was that even "unqualified users" could get into these computers. These people were known as "hackers," and they break into computers just for the sake of finding out what's inside. And by using common passwords–and sometimes randomly chosen code numbers–these hackers could get into banks, hospital records, and even federal agency computers.

Cheifet then provided narration over some footage from *WarGames*, noting that in the movie, adolescent hackers gained entry to top secret defense programs. Cheifet said experts claimed the possibility of this happening in real life was essentially zero. Nevertheless, federal officials were planning to strengthen existing security procedures. Cheifet said there were also professional computer hackers–a "new kind of spy" who traded in the trench coat and camera for a keyboard and a video display. A growing number of institutions were figuring out how to foil this new breed of thief. Ultimately, Cheifet said the question was, "Can a computer really be programmed to keep its secrets secret?"

## The Escalation of Computer Crimes

Donn B. Parker of SRI International joined Cheifet and Kildall back in the studio. Parker was considered the country's "foremost expert in computer crime," as he'd been working in the field for 30 years. Kildall opened the discussion by asking Parker to talk about the scope of computer crime. Parker said that we did not have any firm statistics on the problem, so researchers like himself had to rely on a case-by-case analysis. He noted that in just the past four years, we'd seen a number of record financial losses attributed to computer crimes, including:

- The largest funds transfer fraud, $10.2 million, a reference to the Security Pacific National Bank heist of 1978, where a 32-year-old computer expert, Stanley Mark Rifkin, learned the bank's secret code numbers for wire transfers and posed as a bank official to transfer the money to a Swiss bank as part of a scheme to purchase and smuggle diamonds. (Ironically, Security Pacific ended up selling the recovered diamonds for a profit.) Rifkin later received an eight-year federal prison sentence.
- The largest bank embezzlement, $21.3 million, a reference to boxing promoter Harold Rossfields Smith, who concocted what the *Washington Post* described as "an elaborate scheme" to siphon the money from Wells Fargo "to become one of boxing's most flamboyant and free-spending promoters." A federal jury convicted Smith on 27 criminal charges. He received a 10-year prison sentence, served a little over 5 years, and later returned to boxing as a promoter.
- The largest securities fraud, $53 million.
- The largest commodities fraud, $50 million.
- The largest inventory fraud, $67 million, a reference to the Saxon Industries Inc. bankruptcy case of 1982, where officials at a photocopier company used false financial data to inflate the amount of parts they actually had by nearly double.

Parker said these stories illustrated that the more people used computers, the larger the potential losses from computer crimes were. Kildall asked Parker if that meant he saw "escalation" as the real problem. Parker said yes, noting that while overall business crime was probably going down to increased computer usage, the losses that did occur now tended to be significantly larger.

Kildall asked Parker to describe the “characteristics of a computer criminal.” Parker said his research team had interviewed about 35 hackers to date, individuals he described as among the more “sophisticated computer criminals.” He noted they tended to be young, which was not a surprise given that people in the computer field were generally young. Parker said these individuals tended to exhibit a variation of “Robin Hood” syndrome–i.e., they stold from the rich but kept the money for themselves. At the same time, these hackers did not view themselves as criminals. They “very strongly” differentiated the idea of doing harm to people, which was immoral, with rationalizing the harm they did to organizations.

One such rationalization, Parker said, was that hackers saw their victims as the computers themselves. And a computer cannot “cry or hit back.” That made it an ideal target for someone who would not otherwise walk up to a person, stick a gun in their face, and steal their money. Such behavior was clearly criminal. But things were different when money was stolen through a computer terminal. Parker said hackers ultimately considered themselves “problem solvers.” That is, they had some “intense, unshareable problem they are trying to solve.” This led them to abuse a position of trust they had, as that was the easiest way to solve their problem.

Cheifet asked Parker to clarify if we were talking about “amateurs” or “career computer criminals.” Parker said so far, the problem was mostly amateur, white collar criminals. But we were starting to see an increasing number of career criminals involved with computers. He added such an increase was to be expected given that “for several years now, every major prison in the United States has been teaching data processing to prisoners.” As a result, many career criminals have found they are unable to commit “traditional crimes” outside of the computer environment.

Kildall asked about some of the specific techniques used to attack criminal systems. Parker noted that SRI had researched about 1,100 specific cases over the past 13 years, which led them to classify a number of techniques. He said the most common was *data diddling*, or altering data as it is entered into a computer system. Some of the other common techniques he listed included *Trojan horses*, *logic bombs*, *salami attacks*, *piggybacking*, *data leakage*, and *super zapping*.

At Kildall’s prompting, Parker explained one of the terms, *Trojan horses*, as an example. This referred to building secret instructions into a computer program, so when it is later executed on a system, it not only performed what it was supposed to do but also the additional instructions. Kildall said such instructions could be used to transfer money into someone’s bank account. Parker replied a Trojan horse could be used “for a whole variety of programmed frauds.” Another example was a *logic bomb*, where the program contained instructions that were triggered when certain conditions were met.

Kildall asked about the hypothetical scenario presented by *WarGames*, where a system was attacked from the outside by someone guessing passwords to gain entry. Parker said there were a “whole range of malicious system hacking techniques” to impersonate authorized users of computer systems via a remote terminal. This required knowledge of an authorized user’s ID and password. It was also possible to automatically or manually scan a list of telephone numbers or possible access codes.

Kildall asked if the advent of more “user-friendly” computer systems have helped or hindered computer crime? Parker observed that a friendly system was “almost of the opposite” of a secure one. So the trick was to “find the balance” between the two. As the number of people using computers continued to grow, the enemy was also expanding in size and sophistication. This meant that in some respects, we could “no longer allow systems to be as friendly as they have been.”

Kildall turned to the question of computer bulletin board systems, specifically so-called “pirate” boards used by hackers. Parker explained there were about 128 known pirate bulletin boards in use throughout the country. They were used primarily for “intelligence purposes” by “malicious systems hackers.” For example, pirate boards often broadcast the telephone numbers of computers and passwords that could be used to gain unauthorized access.

## Promoting Basic Password Security

In the final segment, James L. Holmes of Tri-Data Corporation joined Cheifet, Kildall, and Parker, to demonstrate his company's Digi-Link product. This device was designed to help secure dial-up computer networks. It basically consisted of a 212A modem with "answer verification" capabilities. Essentially, the Digi-Link would only give someone trying to access the protected system remotely one or two chances to correctly enter a password before automatically disconnecting the call. This was necessary as many systems of the time allowed an unlimited number of login attempts.

Holmes then explained the use of passwords to secure computer systens. He noted that the Digi-Link could accept passwords of between 1 and 250 characters. The typical password, however, was only about 5 or 6 characters. Obviously, the longer the password, the more difficult it was to break in, and many people who were "intent" on security would try to use passwords of 20 to 30 characters.

In response to a question from Kildall, Holmes said the Digi-Link did not dial back a user to confirm their identity, it only checked for a valid password to log on to the protected system. Cheifet noted there were other systems besides the Digi-Link that used what he called a "Pizza defense," i.e., the pizza parlor calls you back to confirm you are the person who actually ordered the pizza. Similarly, these devices call the dialer back to make sure it was a "proper party" who was requesting services.

Kildall asked Donn Parker if devices like the Digi-Link adequately secured data. Parker said they added another layer of security and thus helped to provide "a significantly more secure system today" than existed in the past. That said, no computer system commercially available today was "adequately secure" relative to the value of the information it stored. It was therefore important to compensate for the technological limitations of systems through operational, physical, and procedural methods to make the *organization* itself more secure.

Cheifet pointed out that the American computer industry often looked to Japan when it came to the latest technology. He asked Parker if there was anything the Japanese were doing in terms of security that we could learn from. Parker said the Japanese were actually a "little bit behind" on this subject as they had not yet experienced the types of computer crimes seen in the United States. That was starting to change, however, and given the eagerness of Japanese manufacturers to enter the American market, they needed to have a stronger understanding of computer crime. Parker noted, “Crime is a very cultural thing,” so its understanding varied from one country to another.

On that point, Cheifet asked if the laws in the United States were adequate to deal with computer crimes. Parker said the laws were “improving a great deal” in this area, although there was still a long way to go. He noted there were now 21 states that had specific computer crimes laws, as well as several bills pending in Congress to provide “specific protection on a nationwide basis.”

Kildall asked about the role operating systems played in computer security. He noted that something like UNIX had source code that was now available to individuals. Could such access lead to security concerns? Parker quipped that with a big enough hammer, you could break anything. So any computer system could be broken with a sufficient amount of work, skills, knowledge, and access. In terms of long-term research on computer security, Parker said the ultimate goal was to develop a “provably secure system,” where you could prove how the computer would perform “under all conditions.”

Cheifet asked about the potential risks of computer crimes with respect to banks, noting they were now all tied to the electronic funds transfer system. What was the potential for a “major economic disaster” if that system were compromised. Parker said the potential was “catastrophic,” but overall the risk was “relatively low.” The key was making sure that security–the “electronic fences” protecting the bank–kept up with the developing technology.

Finally, Cheifet asked about the potential for cryptography as a solution to the problem of computer crime. Parker said cryptography was “the most powerful safeguard” ever developed to protect data through communications lines. At the present time there was “no problem to solve” in this area, i.e., there was no risk of “eavesdropping” by third parties. In the future, however, he said this would be a much more significant issue.

## Writing the Book(s) on Computer Crime

There’s not much to say about James Holmes or Tri-Data. I could not locate much information about either. According to a 1988 article in *Network World*, Tri-Data made “modems, processors, and other support equipment.” As best I can tell, the company was in business from 1967 to around 1992.

Donn Parker, then and now, is one of the country’s top experts on computer crime. He literally wrote the book–several actually–on the subject. Parker began working with computers in the 1950s as a programmer on the UNIVAC 1103. After spending the early part of his career at General Dynamics and Control Data Corporation, Parker moved to SRI in 1969, where he remained until his retirement in 1997. Parker initially worked in management at SRI but shifted in the early 1970s towards researching the nascent field of computer crimes.

Parker told the University of Minnesota’s Jeffrey R. Yost in a 2003 interview that he got into computer crime research due to his interest in professional ethics. A “dedicated Christian,” Parker said his early research focused on “ethical conflicts in computer science,” such as the concept of owning computer software. In 1976, Parker authored *Crime by Computer*, widely considered the first definitive book on the subject. He went on to author six more books on computer crime, as well as the Criminal Justice Resource Manual for the U.S. Department of Justice.

## Notes from the Random Access File

- This episode is dated March 26, 1984, and available on the Internet Archive.
- One of Donn Parker’s SRI colleagues, Peter Schwartz, was actually a creative consultant on *WarGames*. According to a 2008 retrospective in *Wired*, SRI was itself “running computerized war games for the military.”
- Tri-Data Corporation was located on Meade Avenue in Mountain View, California, right across the street from where LinkedIn now has its global headquarters.

- This was another episode included in the *Computer Chronicles Telecourse* series, again featuring interstitial segments hosted by Herbert Lechner, who also worked for SRI International.

# Chapter 8: The Hero-1 and the TeachMover

In a recent essay for the socialist journal *Current Affairs*, Matthew James Seidel recounted a story from 2013 where "delivery drivers came up with an unexpected way to prevent robots from taking their jobs. They beat the robots with baseball bats and stabbed them in their 'faces.'" Seidel quipped that "[s]ome robots got off easy; they were merely abducted and shut away in basements."

The intellectual–and sometimes physical–battle over the use of robots to replace human labor was the subject of a late 1983 episode of *The Computer Chronicles*. This program featured demonstrations of two early attempts at making robotics more accessible to students and programmers. There was also a surprisingly in-depth discussion over the long-term implications that robotics would have on society.

## Would the Year 2000 See a Robotic Dystopia?

Stewart Cheifet and Gary Kildall opened the program by playing with a popular toy robot of the day, the Armatron distributed by Radio Shack. Cheifet noted this was a small example of a human-operated robot that did not require a computer. Cheifet asked Kildall to explain how computers and robotics related to one another. Kildall said that robots covered a wide range of uses. For example, a simple "show robot" required a human operator to communicate with the device to perform basic tasks like handing out brochures. But as we added "more intelligence" to robots, that's where computers came into play. The computer could control the robot and make on-board decisions based on its own intelligence.

Cheifet then introduced our B-roll segment for the week, which focused on the portrayal of robotics in popular culture. During the early 20th century, Cheifet said the robot had become "a frightening symbol of the future, threatening to transform the world into an over-mechanized, dehumanized society.” He pointed to the 1927 German movie *Metropolis* directed by Fritz Lang. Cheifet said that film depicted the year 2000, where the world was divided between industrial barons and workers, the latter of whom were slaves to an underground machine city. The drive for efficiency led the city’s master to consult an inventor who devised the “ultimate tool” in the form of a mechanical replacement for the imperfect human being, i.e., the robot.

Cheifet said *Metropolis* may not have been completely accurate in its predictions, but its concept of robots–a human-made, human-like mechanism that behaved like its creator–persisted for decades in the popular imagination. Fear of robots destroying their creator even prompted Isaac Asimov to write his famous Three Laws of Robotics in 1942.

Asimov’s rules, however, assumed that robots possessed either a large degree of intelligence or programmed behavior. Cheifet said the latter was “one of the principal criteria for modern robotic systems.” And while present-day designs did not emulate human characteristics, since the 1960s robots have become “useful adjuncts to a number of industries, particularly in jobs that were uncomfortable or hazardous to humans.” In that sense, robots had started to fulfill the role predicted by *Metropolis*, especially since newer robots had “human-like capabilities including sight, touch, and error detection.”

## The Need for Robots to Take “Dirty Jobs”

Dr. David Nitzan of SRI International and Matt Gerrieri of Autobotics, Inc., joined Cheifet and Kildall for an extended discussion of the ideas introduced during the prior segment. Kildall opened by joking that most people had the idea that a robot should “go around and wash your dishes and clean your house,” but there was more to robotics than that. Nitzan explained that industry had already been using robotics to do “dirty jobs” for the better part of 15 years.

Cheifet asked for clarification: What did Nitzan mean by “dirty jobs”? Nitzan replied that he meant “undesired” jobs that were “hard, strenuous, dangerous, harmful, and lethal” to humans. Kildall asked for an example of a “lethal” job. Nitzan said he was speaking in future terms, but he meant something like working in a nuclear power plant.

Kildall then asked about the use of robots on industrial assembly lines, which was now common in Japan. Gerrieri said industrial robots performed “dehumanizingly dull jobs,” which American workers had “demonstrated a strong dislike for.” Rather than sending production offshore, American industry could fulfill these “mundane assembly tasks” by using robots, which would then “free up human beings for more creative, more challenging kinds of work.” Nitzan added that many assembly line workers, particularly women, were underpaid. This made it difficult for robots to compete with cheap labor, especially when it was outsourced to Mexico or east Asia. On the other hand, when it came to more expensive workers performing more skilled jobs, such as die casting or arc welding, replacing these individuals with robots was more cost effective.

Gerrieri suggested the key was to place an industrial robot “into a structured work environment” where the work was presented in such a way that it did not require “extreme accuracy or sensory capability,” which was typical of more anthropomorphic robots. Gerrieri cited Henry Ford’s innovative use of the assembly line to “bring the work to the worker.” Similarly, the trend in robotics would become structuring the workplace to suit the robot.

Nitzan added that while people worried about unemployment caused by robotics, the actual number of blue-collar workers displaced by industrial robots was “very very small.” And projections for the next 5 to 10 years suggested the impact would continue to be minimal. The main reason for this was that modern robots lacked intelligence, rendering them unable to “compete effectively with human workers.”

Kildall asked if that might not start to change with the growth of artificial intelligence? Gerrieri said there was a spectrum ranging from very simple programs–which involved the types of robot-like devices used in Japan–to “very esoteric” applications that required artificial intelligence and sensory capabilities. But there was also a broad range of applications in the middle that allowed a robot to be cost effective and competitive with a minimum amount of intelligence. Ultimately, Gerrieri said the future was intelligent machines and “distributed intelligence architectures,” where robotic systems would operate within a “totally automated factory” governed by several levels of computer hierarchy.

Cheifet closed the segment by asking Gerrieri where he saw the state of Japanese robotics relative to the United States. Gerrieri said the Japanese were ahead in terms of automating their factories. Japanese manufacturers adopted a systemic, bottom-end up approach to automation, starting with the “easiest applications that they can resolve, learning in the process, and then moving on to the next-most difficult application.” On the other hand, American robotics was ahead when it came to technologies associated with anthropomorphic robots, such a vision systems. That said, robotics was becoming a global industry, as illustrated by Japanese robotics company Fanuc’s recently announced joint venture with General Motors.

## Slowly Adding “Smarts” to Robots

The final segment featured in-studio demonstrations of two commercially available teaching robots. The first was the Hero-1 robot manufactured by Heath (also known as Heathkit). George Oliver, the northern California distributor of the Hero-1, conducted the demonstration for Stewart Cheifet. Oliver began by showing off the Hero-1’s voice synthesis capability to “speak” to Cheifet. Then the robot moved around on its base and extended its arm. Cheifet asked how much weight could the arm hold. Oliver said about 8 ounces while extended and up to 1 pound when fully retracted. The Hero-1 was therefore capable of picking up small items on an assembly line and moving them to another line.

Cheifet asked if the Hero-1 had the same freedom of movement as an industrial robot. Oliver said yes, the Hero-1 was considered a Class 3 industrial robot, and as such had six axes of motion. The robot then demonstrated its ability to move around on its built-in wheels.

Next, Cheifet asked about the Hero-1’s included keyboard and LCD display. Oliver said the keypad was used for data entry and the display showed the user where information was stored. This enabled the user to program the Hero-1 directly.

Oliver and Cheifet then removed the panels surrounding the Hero-1's base to show the circuit boards inside. The main board included the built-in microprocessor–a Motorola 6808–and various integrated circuits. The side boards contained various sensors, such as a sonar transmitter and receiver, as well as the previously demonstrated voice synthesizer. Cheifet noted the Hero-1 was designed to be used "mainly as a training and educational robot." Oliver replied yes, it was used primarily by students working in the laboratory.

Cheifet then rejoined Gary Kildall and David Nitzan, along with Guy W. "Dusty" Rhodes and Dr. John W. Hill from Microbotics. Rhodes and Hill demonstrated the TeachMover, a tabletop robot used to help train industrial engineers. The demonstration consisted of a "block stacking" program. The TeachMover found a large block placed on a tabletop grid and moved it to a pre-prorgammed location. Rhodes explained the robot had sensors to determine how large a block was. The TeachMover also knew to return to its previous location to look for additional blocks. If there were no more blocks, it would return to its starting position and "sit down."

Kildall noted the TeachMover's sensing mechanism did not include any actual vision, i.e., a camera. He asked if such a feature would be added in the future. Hill said it was possible, adding that future additions could include range and touch sensors. Combined with appropriate "smarts" in the form of artificial intelligence, and the TeachMover could not only "know what it is doing," it could "blow the whistle" if it made a mistake.

Cheifet followed up, asking where things stood today in terms of adding "smarts" to robots. David Nitzan answered bluntly, "We are struggling." He said probe sensors were being "gradually" added to robots. This would eventually make robots more useful in unstructured environments, such as agriculture, the military, and medical institutions. He added that "home use" of robots was also difficult today because the machines were "too dumb" to be of any practical use.

Cheifet asked Nitzan if he expected the field of "personal robots" to grow in the same manner as personal computers. Nitzan said he was betting on growth in application-oriented robots. Individual hobbyists were not where the money was right now, except as toys.

Kildall pointed out that the personal computer market had grown due to grassroots efforts, such as the invention of the Apple II by Steve Wozniak when he was still basically a hobbyist. Kildall said robotics would benefit from similar grassroots efforts. Nitzan agreed there were parallels. He noted that computers started out as something only useful in commercial and scientific applications; now they were everywhere. He said the same thing could eventually happen with robots.

Cheifet brought the conversation to a close by returning to the question of replacing workers with robots. He noted the robots demonstrated today were basically just mechanical arms. But could robots eventually replace not just blue-collar work, but also white-collar jobs? Nitzan said it could be done once humans were able to transfer their knowledge to an automated system. For example, a computer-aided design database could include all of the information necessary to perform a certain job. This would eliminate the need for a human trainer to move the robot around by issuing commands via a control panel. Computers would enable the robot to self-teach and self-calibrate in order to perform its tasks.

Cheifet quipped to Kildall if they weren’t replaced by robots, they’d be back with another episode next week.

## Computer News from the 1983 Holiday Season

This is the first *Chronicles* episode, at least in the order I’ve been reviewing them, to include the post-show “Random Access” segment, which was a rundown of computer-related news headlines. Stewart Cheifet presented this segment initially, with other presenters taking over in later years.

Here were the stories from this episode’s “Random Access,” which aired just before Christmas 1983:

- While Cabbage Patch dolls were the best-selling new toy that holiday season, the “real hot Christmas gift” was the low-priced home computer. Cheifet said analysts estimated there would be 2.5 million computers under the Christmas tree in 1983, overtaking video game consoles.

- Cheifet added that there was also a growing supply of used computers, which drove demand for computer swap meets and used computer stores. Two examples were Palo Alto-based Computer Swap America, which planned to expand to five additional cities in 1984, and the Interstate Computer Bank, which previously sold used computers by mail order but was now planning to open a retail location in Mountain View, California.
- Meanwhile, due to the Great Video Game Crash of 1983, Atari and Activision looked to improve their fortune by announcing a joint venture to distribute games via television; they planned to test the new system in early 1984.
- Cheifet said 1984 was expected to be the "Year of the Mouse," with sales of the input devices growing to $10 million in 1984; optical mice were seen as likely to supplant traditional mechanical devices.
- Semiconductor manufacturers also expected a record year in 1984. Domestic orders for computer chips hit an all-time high of $1 billion in November 1983, and industry analysts expected sales to grow another 30 percent in 1984.
- Chip sales also drove semiconductor stocks to record highs, with Intel hitting $41 per share in December 1983. Other companies were not doing so well, including Apple, which had sunk to $25 per share after a 1983 high of $63.
- A company called Value Line–I may not have the spelling correct on this one–announced plans to start a monthly software service for picking stocks. Each month, Value Line would send subscribers a floppy disk that measured "32 factors for some 1,600 common stocks" and provide recommendations.
- Osborne Computer Corporation emerged from bankruptcy under a new chief executive, Ronald Brown, who planned to "get out of" the manufacturing and domestic computer markets and focus on making IBM PC compatibles.
- Speaking of IBM, there were rumors about looming upgrades to the company's struggling IBM PCjr computer, notably an upgraded keyboard following "lukewarm press" for the original Chiclet-style keyboard.
- In more Apple news, the company "finally made some progress" in its battle against unlicensed Apple II clones. United States customs

officials seized 400 “fake Apples” from computer dealers in the San Francisco area.
- The original creators of the BASIC programming language, John G. Kemeny and Thomas E. Kurtz, announced plans to release a new language called True BASIC in 1984.
- There was a new world record for largest number factored by a computer. A Cray supercomputer factored a 67-digit number in 13 hours.
- Two students in Sacramento, California, shot a computer used to keep attendance records.
- A robot “testified” at a hearing of the New Jersey Casino Control Commission. A casino owner wanted to use the robot to promote entertainment shows on the casino floor. Cheifet said Commission members noted the robot was only 1 year old and “minors were not allowed on the gaming floor.”

# David Nitzan (1925 - 2010)

In terms of this week’s guests, I don’t have much information. Dr. David Nitzan passed away in 2010 at the age of 84. According to his obituary, he lived in Palo Alto for more than half a century. He was born in 1925 in Jerusalem and served as an intelligence officer in the Israeli military before coming to the United States to earn his doctorate in engineering at the University of California, Berkeley. Nitzan worked at SRI International from 1959 until his “semi-retirement” in 2002, spending most of his tenure directing the institute’s robotics program.

Dr. John W. Hill of Microbot also did a stint at SRI, where he worked on robotics systems for NASA’s Jet Propulsion Laboratory. As for Microbot, it relocated from Silicon Valley to Michigan sometime around 1985 and was later sold to a British company, Oxford Intelligent Machines.

## The “Low Cost” Robots of the Early 1980s

So let’s talk about the robots. As noted earlier, the Hero-1 was produced by the Heath Company, which is still in business today. The Hero-1 was the first in a series of robotics “kits” sold by the company during the 1980s.

The January 1983 issue of *Byte* magazine featured a detailed article on the Hero-1 written by Steven Leninger, who described the unit as a "distant cousin of R2D2." While "not as strong, fast, or accurate as its industrial counterparts," Leninger said the Hero-1 nevertheless featured an "impressive list of capabilities," including the ability to "sense sound, light, motion, distance, and time." The robot's rechargeable batteries lasted for about one hour of "untethered operation" but could also be plugged in with a power adapter.

The basic Hero-1 kit, which required assembly and did not include the arm or speech synthesizer demonstrated during the *Chronicles* segment, retailed for $1,000. The arm would set you back an additional $400 and the speech synthesizer another $150. You could also buy a kit with everything for $1,500–saving a whopping $50–or for $2,500, you could get a fully assembled Hero-1 robot.

For $2,500 you could also get Microbot's TeachMover, which already came assembled and actually retailed for $2,395, according to an advertisement in the July/August 1982 issue of *Robotics Age*, which also featured a write-up on the machine from John W. Hill and one of his Microbot colleagues, Clement M. Smith. They pointed out the TeachMover offered a "low-cost tabletop robotic arm" that did not require an external computer to program or operate, as it came with a built-in 6502A microprocessor, the same 8-bit processor that powered the Apple II and the Commodore PET home computers. And while $2,395 may not sound low-cost even by today's standards, Hill and Smith noted that "control boxes used on large industrial robots" cost around $40,000 in late 1982.

From what I could tell, the TeachMover was actively used in engineering classrooms up and until fairly recently. For example, I found this 2018 undergraduate project from Georgia Gwinnett College where students used modern Python to program a TeachMover to play chess.

## Notes from the Random Access File

- Okay, the more I delve into *Chronicles* history, the more confusing the airdate situation becomes. In my initial posts, I relied on a list of airdates published at TheTVDB.com. But these dates do not match the ones on the episodes published at the Internet Archive. And those dates also seem to be off. For instance, the Archive dates this episode as April 2, 1984. But the "Random Access" segment was clearly from December 1983.

- Regardless of when it first aired, you can watch this episode at the Internet Archive.
- The Radio Shack Armatron featured in the opening segment was the first *Chronicles*-demonstrated product that I actually owned as a child. It was way too easy to break that thing.
- Speaking of Radio Shack, there must have been one located down the street from KCSM, as this was like the third or fourth episode so far that's opened with Stewart Cheifet demonstrating one of its products.
- I'm not sure why Matt Guerrieri was a guest on this episode. He was introduced as the vice president of sales and marketing for Autobotics, yet he did not demonstrate any products or explain what his company did. I wonder if there was a product demo that was planned but scuttled just before taping.
- In case you were wondering, had you purchased $1,000 worth of Apple stock at its December 1983 low, it would be worth roughly $180,000 today.

# Chapter 9: The VOTAN V5000 and the Speech Plus CallText

This next episode of *The Computer Chronicles* from early 1984 examined the status of speech synthesis technology. Stewart Cheifet opened the program by showing off a mini version of the popular Speak & Spell toy. He also demonstrated the Minolta AF-S V, a "talking" camera that could vocalize basic warning messages to the user, such as that lighting conditions were "too dark!" and the operator needed to "use flash!"

Cheifet was joined in the co-host's chair this week by Herbert Lechner of SRI International. With respect to the Minolta, Cheifet noted the talking feature wasn't strictly necessary; a blinking light worked fine for most cameras. So he asked Lechner if the use of speech in such devices was a "marketing gimmick" or if there were useful applications for speech in computers. Lechner replied that he found himself not paying attention to the warning lights in his car, yet he paid attention when it talked to him. And there were a number of voice terminals in use today that relied on telephone technology.

Cheifet then transitioned into our B-roll feature for the week, which provided an overview of the research into not just speech synthesis, but also digital speech and speech recognition. There was footage of a person operating an electron microscope. This device required total darkness to operate due to the machine's sensitivity to light. In addition, the operator's eyes and hands were kept continually busy simply dealing with the microscope. So the attached computer was trained to "respond to verbal commands encoded previously by the operator in his own voice."

Cheifet said machines that could understand and carry out such verbal commands had been in development since the late 1960s. And even in the mid-1980s, the applications of such technologies were still limited. He said the "breakthrough" would come with the perfection of more recently developed speech systems that recognized context-sensitive rules of language and could mimic the more flexible patterns of actual human speech.

## Massaging the Phonemes

Carl Berney, a vice president with Speech Plus, Inc., joined Cheifet and Lechner for the next segment. Lechner opened the discussion by asking Berney to explain how a computer synthesized speech. Berney said his company's product–a Speech Plus CallText synthesizer connected to an Epson HX-20 computer–could take any ASCII text and convert it into intelligible speech. The circuitry first looked at the words, decided which phonemes (the basic elements of speech) were in those words, and then put those phonemes together. The machine also "massaged" the words to make sure that they flow in the same way as natural speech. This was then converted to a set of parameters for the speech synthesizer, which produced the final speech.

Cheifet asked Berney to explain phonemes further. Berney said a phoneme was basically a "short sound." It was a fundamental element of language such that changing a phoneme would also change the meaning of a word. He said there were roughly 38 to 45 phonemes in the English language depending on which linguist you asked. Other languages had roughly the same number.

Lechner asked if the CallText covered the entire English language. Berney replied that it did. Lechner asked what type of computer was necessary for the speech synthesizer to work. Berney said the product was based on a commercially available 16-bit microcomputer.

Cheifet asked what would happen if you typed French words into the CallText. Berney said the device would attempt to pronounce the words, but they would not come out right, because French uses a different phoneme model than English. Cheifet said that meant the CallText was language dependent. Berney said that was correct, noting that it was not just the phonemes that differed between languages, but also the rules of pronunciation. The CallText contained pronunciation rules specific to

English. This helped the final speech come out with a "reasonably natural inflection."

Cheifet asked for clarification on what it meant to "massage" the phonemes. Berney explained that when a person speaks, they did not normally articulate everything. For example, if you were giving directions and said, "Make a left turn at the corner," you might not clearly articulate "left" and "turn" separately; the words would probably run together. Put another way, some words are strung together in normal speech and some sounds are not spoken at all in certain words. There were rules for how to adjust a phoneme depending on which sound came first and which sounds came afterwards. It was therefore necessary to know when a sound appeared within a word (and a sentence) to get the proper inflection.

Lechner followed up, asking if the manipulation of the phonemes included volume and inflection. Berney said yes, the phoneme was not a fixed element. It needed to flow to another point depending on what sound came next.

Cheifet noted that during the introduction, he and Lechner described this technology in terms of "speech synthesis." But that was different than "speech coding." Could Berney explain the latter? Berney said the CallText did "synthesis by rule." The CallText did not use any pre-recorded or stored speech. Instead, it took written text and produced speech based on its understanding of the pronunciation rules of English. Other speech devices relied on "synthesis by analysis," where the information was recorded from a human voice and then compressed or processed by a computer so it could be stored and reconstructed later. These devices had "no knowledge of what the words were" and thus no idea about how they flowed together. It was essentially a "sophisticated tape recording process."

Berney then demonstrated the CallText on the HX-20. He had stored some text messages ahead of time, which the device played. Cheifet asked if it were possible for him to call up his computer remotely and have the CallText read back his electronic mail messages to him. Bernery said it could. The CallText could access "any text database over the telephone." Lechner asked how big was the market for such a feature. Berney replied it was a new application, and as of now there were no commercial online services that supported the CallText. However, Speech Plus was talking to “most of the major electronic mail suppliers” about potential adoption.

Cheifet then asked Berney to show the actual CallText product. Berney explained it was an expansion board that plugged into an IBM PC. The board had its own microprocessor and both a standard telephone interface and an RS-232 serial port.

## An Early Voice Menu System

For the final segment, Rodney Stephens of VOTAN joined Cheifet, Lechner, and Berney. Stephens demonstrated the VOTAN V5000. This device could control a personal computer using voice recognition. Like the CallText, it connected to an IBM PC using an RS-232 connection. The included software package controlled a color monitor.

Basically, the demonstration consisted of Stephens talking into a microphone and using simple verbal commands to navigate menus within a sample program that displayed sales figures for a company. Stephens also demonstrated an application that enabled the user to verbally access their brokerage account, check their portfolio, and even leave a voicemail message for their stockbroker.

Cheifet remarked the V5000’s speech was so good, it was hard to believe that it was computer generated. Stephens said the V5000 relied on “analysis and coding” technology that was similar to that used by major telecommunications networks. The main difference was that those networks used 64,000 bits of speech per second to modulate digital speech, while the V5000 only used about 10,000 bits per second. This made his device more cost effective and feasible for use with a personal computer.

Lechner observed the V5000 relied on a pre-recorded and digitized vocabulary that was then sampled at 10,000 bits per second. Stephens said yes, the V5000 stored about 1,000 messages altogether that could be accessed by the included software, which could call out one message at a time. It was also possible to combine several messages together.

Lechner then asked about where “the state of the art” was with respect to speech recognition technology. Stephens replied there were several factors to consider. First, the V5000 relied on speaker-dependent recognition using discrete words. Speaker-independent recognition was a more difficult technology and it would offer a more limited vocabulary. Second, there was the problem of discrete versus continuous words. Stephens said we would not see speaker-independent recognition of continuous words anytime soon. Finally, there was the scale of the vocabulary. The technology for accomplishing unlimited vocabulary with speech recognition–something akin to what the CallText could do with speech synthesis now–was probably 2 to 4 years away.

## Hawking Makes the CallText Famous

I could not find much about VOTAN or its president, Rodney Stephens. The company existed from 1979 until 1991. And according to one source, the V5000 retailed for $5,000.

As for Speech Plus, its CallText line of speech synthesizers became famous for its association with the late Professor Stephen Hawking. In a 2015 article for *Wired*, writer John Medeiros explained that when Hawking lost his ability to speak in 1985 after contracting pneumonia, he started using a computer program called Equalizer to type out words and commands, which were then synthesized into speech by a Speech Plus CallText card attached to an Apple II. The use of the American-produced card was why Hawking’s famous digital voice carried an American accent. Speech Plus later gave Hawking an updated model–the CallText 5010–in 1988, but he asked to keep his “original” voice, as he’d become accustomed to it.

Medeiros noted that years later, one of Hawking’s graduate assistants wanted to make a software-based version of the professor’s voice so that they would not have to “rely on these old hardware cards” anymore. That required finding the original Speech Plus team. The company itself was long gone. Centigram Communications purchased Speech Plus in 1990. Centigram itself then went through several rounds of corporate acquisitions. Eventually, the graduate assistant managed to track down a backup take of Hawking’s original synthesized voice at Nuance Communications.

### Berney Makes His Mark with Software and Sculpture

Carl Berney, the Speech Plus executive who demonstrated the CallText on *Chronicles*, went on to have an interesting career both inside and outside of technology. Berney worked at Speech Plus, and later Centigram, for more than two decades. During his tenure, he managed the development of Voice Gateway, a widely used integrated telephone voicemail and response system. Since 2014, Berney has been involved in mobile software development, publishing the app ShotPut, which helps diabetics rotate the sites of their insulin injections.

Berney has also had a prolific career as a stone carver. He owns and operates Standing Stone Studios in San Luis Obispo, California, where he produces artistic works based on marble, limestone, and alabaster. Berney also spent three years as a sculpture instructor with the Poeh Cultural Center in New Mexico.

## Notes from the Random Access File

- This episode is available on the Internet Archive, which provides a date of April 9, 1984.
- There was a *lot* of Herb Lechner in this episode, as he also presented a number of interstitial segments from the *Computer Chronicles Telecourse*.
- Lechner mentioned his car talking to him. Some early 1980s cars could do this, but the technology did not involve any speech synthesis; rather, there was a small phonograph in the dashboard that played short, prerecorded messages.
- We will see Carl Berney again in at least one future *Chronicles* episode.
- The Epson HX-20 used for the CallText demonstration was itself a noteworthy device. It was widely considered the “first laptop computer.” A Japanese import introduced to North America in late 1982, the HX-20 came with 16 kilobytes of onboard RAM (expandable to 32 KB), a 4-line LCD display, and two 8-bit Hitachi 6301 microprocessors that combined to provide basic 16-bit functionality. The machine also had a built-in dot matrix printer.

- A sign of the pre-Internet age: Stewart Cheifet and other participants kept calling it “electronic mail” rather than “e-mail.”
- I almost managed to get through this entire post on speech synthesis technology without remarking how this episode’s demo products probably work better than Siri. Almost.

# Chapter 10: The Sytek LocalNet

Today, we think of networking as synonymous with the Internet–a global interconnected network that encompasses not just computers but also millions of “smart” devices. But in this episode of *The Computer Chronicles* from late 1983, the focus was on *local area networking* or LANs. Stewart Cheifet and Gary Kildall talked with representatives from two companies that were at the forefront of developing the still-emerging standards for computer networking.

Cheifet opened by asking Kildall to define a local area network. Kildall noted that ever since we’d had computers, people had beeen trying to hook them up to transmit data back and forth between them. LANs were generally used in office environments and limited to a relatively small geographic area. This limited area allowed for better performance relative to more traditional network systems.

Cheifet then narrated our B-roll for the episode. He explained that the “countertop” computers–i.e., terminals–typical of modern offices had a built-in limitation: They could only talk to a central computer linking it and other terminals to one body of data. Networking was a relatively new field that allowed an organization or community to connect its various branches into an integrated, decentralized resource. At Stanford University’s central computer facility, for example, a library network gave researchers instant access to material stored all over the country. Individual campus terminals could tap into the system through a yellow cable that “snakes through the campus like a subway line.” Although a LAN can operate along traditional telephone wires, known as twisted pairs, Cheifet noted that newer single-cable systems like *baseband* and *broadband* offered more flexibility, a greater data rate, and could bring the tangle of cables and wires “almost under control.”

Cheifet noted there were a number of ways that information could be shared within a network: Along the arms of a star, around a ring, or through a BUS. Messages traveling on the network could receive automatic priority or use travel-like “tokens” to get from one station to another. But the real advantage of this system was for the user, whose access to multiple sources of information became easier and faster.

## Broadband vs. Baseband

Dr. Charlie Bass and Dr. Michael Pliner joined Cheifet and Kildall for the next segment. Kildall asked Bass, the vice president of Ungermann-Bass, Inc., to explain the characteristics of a LAN. Bass replied that a LAN involved “total connectivity” between devices and “peer-to-peer relationships,” meaning the goal was to get rid of the central computer entirely, thus allowing individual devices to talk freely among themselves.

Kildall asked about the types of networking applications used in office environments with smaller computers. Bass said the main goal was to share resources and information, so that a single terminal can talk to more than one computer, and a user could share resources they might not ordinarily be able to afford. Pliner, the president of Sytek, Inc., added that LANs were now common in office automation environments. For instance, they were used on manufacturing floors to collect and transfer data back to an operations center. LANs were also used for general communications utilities. Indeed, the network itself was now becoming a utility much like electricity. Newer broadband networks could even provide video, voice, and data on the same cable system.

Kildall asked about the differences between broadband and baseband technologies. Pliner said the distinguishing feature was the modulation technique. In a baseband system, you directly modulated the cable with electrical signals. In a broadband system, you are essentially using the same technology that distributes cable television into homes. It uses an RF-modulation technique or frequency division multiplexing, i.e. stacking different channels of data, information, or video that run along the same cable simultaneously.

Kildall asked about the data-transfer rates of these networking technologies. Pliner said it ranged from around 9.2 kb/s to Ethernet-type networks that could get up to 10 MB/s. And newer technologies like broadband and fiber-optics could get even faster.

Cheifet asked how decisions were made regarding which networking technology was best for a particular application. Bass said that much of the media discussion over “broadband vs. baseband” was missing the point. Most people now realized that both technologies had their strengths and applications and would continue to co-exist. This would not be a case where one technology killed off or dominated the other to its exclusion. That said, choosing between the two was difficult. It was a matter of application, cost, and efficiency. Baseband offered simplicity, which resulted in savings not just in terms of cost, but also installation and maintenance times. Broadband was more complex and costly, but if you needed the ability to transfer something other than a simple data stream, that was the only way to go.

Kildall asked about the software issues related to LANs. Did they require a lot of specialized software to operate? And were operating systems doing enough to support networking? Bass said there was an entire body of specialized software used for networking, which were known as *protocols*. This software was necessary for most existing computer systems, as they were built “without any knowledge of the network.” Over time, Bass said he expected you would start to see these protocols permeate the operating systems themselves. That is, future operating systems would be built with the assumption that a network existed. This would allow for networking to incorporate more applications and become more powerful.

Pliner agreed with Bass that as things stood now, the LAN vendor still had to take a lot of the software and “move it into” the computer. He pointed to Sytek’s networking devices, which included its own microprocessor as well as a “significant amount” of communications software. In the future, you would likely see specialized software on individual host machines to provide networking applications such as distributed process, file sharing, and electronic mail.

Cheifet asked about the problem of technical standards in LANs. Was it possible to link hardware and operating systems from different vendors? Pliner said it was possible. Both he and Bass’ companies produced networking equipment that relied on RS-232 connections. This allowed for the interconnection of multiple vendors' equipment at the cost of some limits in performance and function. Pliner said the Institute of Electrical and Electronics Engineers (IEEE) was currently developing future networking standards with the participation of both Sytek and Ungermann-Bass.

Kildall followed up by asking about the importance of standards. Did a standard really matter if you have a LAN that was self-contained in one geographic area? Bass said standards activity was driven by the need to interconnect different devices. The more commonly accepted a standard, the simpler that would be. But since the standards process had to start from scratch, the first step was looking for the "lowest common denominator," which in this case was RS-232. It wasn't ideal for networking, he said, but it existed and was a well-established standard.

Bass added that right now, much of the standards discussion was focused on very low-level issues, such as the types of cables that should be used for LANs. Once those issues were resolved, standards bodies could focus on higher-level problems. And while the process began with the hopes of a single standard emerging, we would probably end up with a few different standards and need to develop gateways to bridge the gap between them.

## A Step Towards Transparency

For the final segment, Sytek engineer Phil Edholm demonstrated the company's LocalNet product line. The first demonstration involved a terminal connected to a small prototype network. As Pliner explained earlier, the LocalNet had its own processor, memory, and software that enabled networking over a cable. Edholm noted this technology could be used to support "thousands of terminals" on a single cable.

Edholm also demonstrated how the LocalNet could be connected directly to an IBM Personal Computer. He connected a demo PC to the LAN and download a file using a local terminal simulation program. Edholm also showed that the same networking cable could carry a video signal–in this case the live feed from the *Chronicles* taping–to a Sony Watchman.

Kildall asked if LANs were limited to file transfers, or could they be used to distribute an entire operating system over the network? Put another way, could a local computer simply access the files remotely without downloading them first? Edholm said that the network communicated with the IBM PC as if it were a terminal. The PC had to initiate the file transfer using a small piece of locally installed software. But, Kildall said, in the long term this operation would become completely transparent. Edholm said yes, that was the goal. Charlie Bass added that with transparency, the commands the user saw would be natural and that they would have "no idea" where the resource was. The file transfer would take place behind the scenes and not because the user explicitly requested it. And such transparency would come with the next generation of devices that included built-in networking capabilities.

Kildall returned to the discussion of standards. He noted that Xerox, Intel, and Digital Equipment Corporation had proposed a networking standard called Ethernet. What was the status of that proposal? Bass said in some terms it was already "very successful." A number of companies had endorsed Ethernet and were making equipment based on that standard. That said, Ethernet was not an end-all in itself. It represented a basic technology. But it was not the only technology available, as there were alternatives such as broadband, which Sytek used.

Bass noted that the problem with standardization up to this point was that everyone was looking for something that would solve all of the problems and they didn't find it. It was similar to the search for a "universal computer"–there wasn't one and we probably would not be happy with it if it existed. Pliner ended the discussion, and the episode, by noting there was a danger that if the industry adopted a networking standard too early, there was the danger of picking the wrong technology. And in any event, the real question going forward was, "What will IBM do?"

## "Now That's Networking!"

This episode was followed by a Random Access segment hosted by Stewart Cheifet, which dates this program sometime in October 1983.

- C.D. Anderson, a discount stockbroker, announced plans to publish the "first home brokerage computer system" that would cost $300. Charles Schwab said it would introduce its own software sometime in 1984.
- A new survey found that 86 percent of U.S. high schools had at least one personal computer available for student use; that figure was 61 percent for elementary schools. Cheifet noted that some educational experts said it was a "waste of time to teach kids programming." Another research firm estimated it would cost $4.5 billion to equip all U.S. schools with computers.
- Another study, this one from Stanford University, suggested boys were more interested than girls in computers. The study noted the vast majority of students enrolled in computer camps were boys, as was the case with computer ownership and computer game purchases. Cheifet said that could "create real problems for women in the workplace of the future."
- IBM and Hitachi settled their lawsuit over the latter allegedly stealing the former's trade secrets. Hitachi agreed to return and not use the stolen information, and to disclose the names of the people who offered to sell the trade secrets in the first place.
- AT&T announced the sale of a $70 million computer software system to Nippon, the Japanese telephone company. Nippon also planed to buy a $12 million Cray supercomputer to run the software.
- In Silicon Valley news, John Couch left Apple after running the company's Lisa division. But Apple gained another executive, luring John Cavalier away from Atari.
- The publishing industry said there were now more than 2,000 computer how-to books in print. Cheifet said that computer books now outsold all other business-related books–a growth matched only by romance novels. In addition, there were now 72 computer magazines on the market, which was remarkable given this market did not exist just six year ago.
- Cheifet gave a rundown of the week's top-selling computer games: *Zaxxon*, *Wizardry*, *Microsoft Flight Simulator*, *Pinball Construction Set*, and *Zork One*.
- Finally, Cheifet relayed the story of what may have been the first successful online romance. Two CompuServe users, known as "SweetLady" and "MrMike", met in the service's CB channel. Cheifet

> said the couple recently got married on CompuServe with 70 guests logged in. Cheifet added, “Now that’s networking!”

# IBM Adopted–Then Abandoned–Sytek

Let’s start with the answer to Michael Pliner’s question, “What will IBM do?” As it turned out, IBM went with Charlie Bass’ Sytek. In 1984, IBM released its first LAN system, the IBM PC Network, which was built on Sytek’s broadband protocols.

But let’s back up a bit and talk about the background of Pliner and Sytek. Pliner earned a doctorate in computer science and began his career in the early 1970s with the National Security Agency. In 1975, he joined Ford Aerospace in California to work on software development. During his time at Ford, Pliner met four other individuals and together the five of them became the founders of Sytek.

In an interview with James Pelkey, Pliner explained that Sytek was originally founded as a technology consulting firm, not a networking company. Their first contract was to write protocols for Xerox’s X10 telecommunications network.

During their early consulting work, Pliner said he began to hear talk about LANs, and broadband technology in particular. Around 1980, Pliner said Sytek decided to take its consulting money–about $300,000 at that point–and reinvest it into developing broadband. Sytek’s initially focued on selling broadband networks to large college campuses. Pliner told Pelkey that after a talk at Brown University in 1983, IBM representatives approached him about adopting Sytek’s broadband to build “metropolitan” networks of IBM Personal Computers. This eventually led IBM to “re-engineer” the Sytek LocalNet into the original IBM PC LAN.

In September 1984, IBM purchased approximately 5 percent of Sytek for $6 million. But IBM also quickly abandoned Sytek and broadband in favor of another technology called token ring networking. By the end of 1985, Sytek was “fifth in sales among LAN vendors,” according to Pelkey. Sytek’s plans for an initial public offering fizzled. The company failed to post any profit until 1988, and it was sold to Hughes Aircraft, a subsidiary of General Motors, in 1989, and re-branded as Hughes LAN Systems. This subsidiary was in turn sold to Whittaker Communications in 1995.

Michael Pliner left Sytek in 1987. A year later, he founded Verity, a software company that focused on document search and distribution. He served as Verity's CEO until 1993 before moving over to KLA-Tencor as vice president and general manager of its software division. In 1997, Pliner founded Vuent, which developed 3-D streaming models for the Internet. After selling Vuent in 2000, Pliner went on to serve as interim CEO for three different companies before founding yet another startup, FusionSeven, in 2011, which provided financial services software-as-a-service.

The other Sytek employee featured in this episode, engineer Phil Edholm, continues to work in the industry as a communications and networking consultant. Edholm remained with Sytek through its acquisition by Hughes. He left the company in 1991. In 1995, he joined Nortel Networks, eventually rising to become chief technology officers of its enterprise business before departing in 2009.

## UB Struggled to Market Vendor-Independent LAN

Ungermann-Bass–also known as UB–was started by Ralph Ungermann and Charlie Bass. Both men previously worked at Zilog, which famously developed the Z80 microprocessor used in many 8-bit computers of the 1980s. According to James Pelkey, Ungermann was "forced out" at Zilog and decided to start a LAN company. He recruited Bass and other disgruntled Zilog employees to form UB in 1979.

While Sytek focused on broadband-based LANs, UB's specialty was Ethernet. As Bass alluded to during his *Chronicles* segments, Ethernet did become an industry standard, specifically the IEEE 802.3 standard that is still in use today. Bass had led the charge for Ethernet at UB and he decided to base the company's early networking products on the Z80, which he was obviously quite familiar with from his Zilog days.

During the 1980s, UB adopted other networking technologies, including broadband and token ring, in an attempt to cement its position as a "vendor-independent" LAN company. Unlike its rival Sytek, UB managed to go public in June 1983. But the company was never very profitable, and following the October 1987 stock market crash, Ungermann went looking for a buyer, which he found in Tandem Computers. Tandem formally acquired UB in February 1988 for $260 million, with Ungermann staying on as a vice president. Tandem was later acquired by Compaq in 1997.

There's limited information on Charlie Bass' post-UB career. Like many former Silicon Valley executives, he formed a venture capital firm, Bass Associates, in the late 1980s. He co-founded at least two other companies, Starlight Networks (a 1990s streaming media software developer) and Socket Mobile, which is still in business today. Bass also spent a number of years as a consulting professor of electrical engineering at Stanford.

# Notes from the Random Access File

- This episode was recorded on September 26, 1983, although it aired sometime in October based on the "Random Access" segment.
- You can watch this episode at the Internet Archive.
- It's never made clear exactly which Sytek device is used in Edholm's demo. Based on the available product brochures from the time, my guess is it was the LocalNet 20/100 Z01 Option.
- When Charlie Bass referred to RS-232 as a "well-established standard," he wasn't kidding. The original RS-232 standard was introduced in 1960, so it was well over 20 years old when this *Chronicles* episode taped.
- Stewart Cheifet mentioned AT&T's sale of software systems to its Japanese counterpart, Nippon. In an interesting historical footnote, AT&T controlled Japan's telephone network during the post-World War II occupation until it was turned over to the state-created monopoly Nippon in 1952.

# Chapter 11: The Am2901C and Am29116

Computer architecture is usually described in terms of *bits*. For instance, we often speak of early personal computers from the late 1970s and early 1980s as *8-bit machines*. In simple terms, this means that the CPUs in these computers could only address 8 bits of data at a time, with each *bit* representing a single binary digit (0 or 1). But even when the first episodes of *The Computer Chronicles* started to air in late 1983, there were already 16-bit processors on the market, such as the Intel 8086, and 32-bit machines had started to become a reality.

In today's featured episode of *Chronicles*, recorded in October 1983, Stewart Cheifet and Herbert Lechner interviewed a single guest, J. Philip Dowling of Advanced Micro Devices Inc. (AMD), about his company's microchip technology, including a popular 4-bit chip then in its fourth generation.

The episode began with Cheifet noting that as processors moved from 8-bit to 16- and 32-bit, the memory capacity of microchips was also increasing. You could now find 16 KB chips and there was talk of 256 KB chips on the horizon. Cheifet asked Lechner how much more stuff could be "jammed" onto these microchips? Lechner said he was embarrassed to admit that when asked the same question 10 years earlier, he thought the technology was approaching its limits, because he assumed that the individual components of computing would continue to be a "few inches or a foot apart." But now that the distance between components had been reduced to microns–one-thousandth of a millimeter–he had been proven wrong. So who knows how far we could go today.

Cheifet then narrated our B-roll for the week, which described the process of turning silicon into microchips. He explained that purified extract silicon was heated until it became molten and a *seed crystal* was then dipped into the melt and slowly drawn out to form a single ingot. This ingot was then sliced into wafers that were polished and microscopically cleaned. These wafers were the foundation of integrated circuits.

The circuits themselves were created using computer-aided design (CAD). Engineers created and edited news designs using a light pen and video screen. A printout of the computer-edited copy was then routinely checked against the hand-drawn design. The final design was converted into a photographic mask and using photolithography, transposed onto the silicon wafer. A chemical wash developed the pattern by dissolving away or etching sections exposed through the mask. As each layer of circuitry was applied to the wafer, it was checked for proper alignment. Cheifet said this step was especially critical as large-scale integration was replaced with very-large–and soon super-large–integration, representing a jump from 10,000 to 100,000 transistors on a single chip.

## Could Bipolar, MOS & CMOS Work Together?

Phil Dowling joined Cheifet and Lechner for the remainder of the program. Lechner opened by asking Dowling to provide a quick overview of where microchip technology was today. Dowling said AMD and other manufacturers could put thousands of transistors–virtually an entire computer–onto a single chip. He reiterated Cheifet's earlier narration about the use of photolithography, which made it possible to pattern images as small as 1 micron. This meant it was possible not just to build microprocessors but also memory chips with densities of between 64,000 and 256,000 bits.

Lechner asked if the technology had been exhausted, i.e., had we come as far as we could? Dowling replied no, the trend of the past decade–where the feature size of individual chips went down 10 percent per year–would continue over the next decade. In other words, the industry would be able to reduce feature sizes below 1 micron.

Cheifet asked Dowling to explain how the technology allowed for microchips this small. Dowling said it used to involve simple contact printing. Photographic images were placed onto a plate and brought into contact with the material. Over the last 10 years, AMD had moved to using projection imaging, which allowed them to not only get “very, very fine” feature sizes but also meant less damage to the silicon wafers, as the mask was never in direct contract. As a result, there were fewer defects in the manufacturing process, which in turn made it more economical to product “very complex circuits.”

Cheifet followed up, asking if there were more quality control problems–i.e., fewer acceptable chips–as you achieved greater and greater scales of integration. Dowling said that as the feature sizes went down, the number of defects that could be captured during the process do increase. This meant it was essential to maintain cleaner conditions during the manufacturing process. But quality control measures continued to evolve along with the technology. This included *redundancy*, i.e., building extra circuitry into some chips so they could “program out” any faulty portions during testing, leaving “reconstructed good chips.”

Lechner asked for more specifics on the process of designing and modifying a microchip. Dowling noted the design process had changed due to advances in the technology. It used to be possible to draw a circuit design on a piece of paper or a graph and translate it directly onto the photographic plates. But these days, when you had 100,000 transistors on a single chip, that was no longer feasible. Instead, you needed CAD to maintain huge files of various geometries that were ultimately translated into a final design.

Cheifet shifted the conversation to a discussion of heat. Specifically, what were the trade-offs involved as chip designs allowed for “faster and faster movement of electrons” in terms of heat generation? Was there a point where this became a problem? Dowling said yes, it was a problem, and one of the solutions was *complimentary metal-oxide semiconductor* or CMOS fabrication. He explained that each logical element on the chip consumed a different amount of power. With *bipolar technology*, each element consumed a large amount of power to obtain very high performance. With another technique, *metal-oxide-silicon transistors* or MOS, there was less power consumed per element. But even MOS was unacceptable when it came to building hundreds of thousands of components onto a chip. That was where CMOS came in, as it did not burn any power during static operation. In other words, the transistors only required power when they were switched on and off. Dowling said this was the key to building more complex chips.

Lechner asked if the three different technologies could be used in the same computer. Dowling replied absolutely, AMD already offered such products. They produced kits that allowed for bipolar, MOS, and CMOS to all work together.

Cheifet asked if microchip technology was getting ahead of our ability to actually apply and use the amount of integration it offered. Dowling said that wasn’t his area of expertise. But based on his own work in designing chips using CAD, he already saw the benefit of substituting a great deal of computational ability for clever software design. And having engineering workstations that could process millions of instructions per second was something he and his colleagues definitely used.

Cheifet then asked about the possibility of 4-megabit memory chips. Dowling noted that IBM had recently announced a prototype for a 512-bit dynamic memory chip (DRAM). And Seimconductor Research Corporation had also proposed a project to develop 4-megabit DRAM. Dowling said he thought this would be feasible to accomplish in the next 4 or 5 years.

Cheifet ended this segment by asking if silicon would continue to be the principal material for microchips going forward? Dowling said while there were some applications, such as LEDs and optical chips, where other materials made sense, he believed that for very-large scale integrated circuits, silicon was the material that would dominate for the “foreseeable future.”

## The Fourth Generation of 4-Bit Microprocessors

In the final segment, Phil Dowling provided examples of two of AMDs microchips–the Am2901C and the Am29116. The Am2901 chip was first produced in 1975. It was a 4-bit-slice arithmetic-logic unit (ALU). The Am29116, in contrast, was a 16-bit processor with roughly 5 times the component complexity of the Am2901.

Dowling pointed out all of the basic components of the chip, including its register file, instruction decoders, memory, and input/output pins. Essentially, the chip took 4 bits of data out of the register file and pumped it through an ALU, which was controlled by the instruction decoder. Cheifet asked for clarification on what the ALU did. Dowling said that was the part of the chip that did the actual work.

The Am2901C was actually the fourth generation of this particular design. Cheifet asked Dowling to explain how the chip was improved each generation. Dowling said first, changes in the photolithography technology meant the lines on the circuit design could get narrower. This made the final design smaller and more economical to build. There were also changes to the underlying bipolar technology, which doubled the operational speed. This meant the fourth-generation Am2901C had seen a 4x performance improvement over the first-generation design.

As has become *Computer Chronicles* tradition by this point, Lechner then asked about the state of U.S. technology relative to the Japanese. Lechner said some observers felt the Japanese were pulling ahead when it came to the design and manufacturing of microchips. Did Dowling agree? Dowling agreed that the Japanese had come a long way. Ten years ago they were not considered a significant factor in the integrated circuit business. But today they were a “very worthy competitor,” particularly with respect to memory chips. The Japanese were well regarded for their excellent manufacturing capabilities and capital formation, which did give them a significant edge of U.S. industry. This had pushed AMD and other American companies to be more innovative in the products they built.

Cheifet followed up by asking about the fifth-generation computer project under development in Japan and how that would affect chip technology moving forward. Dowling said that was also not his area of expertise, but he noted you needed very sophisticated distributed processing to implement a fifth-generation system. Inexpensive arrays of microprocessors were going to be the key to such systems.

There was then a brief, confusing exchange about the number of transistors that could be placed on a single chip. Lechner thought 100,000 was the state of the art. Dowling said that was actually too low, and in fact AMD already had chips with 250,000 components. And there were microprocessors on the market with 500,000 components.

Cheifet pivoted to a discussion of standards, echoing the previous episode's discussion of networking. Dowling said that AMD was participating in the development of network standards, especially at the lower levels of networking protocols. He said if there was not a good standard defined, it would be hard for companies like AMD to get good parts. And given the company invested years of design activity into its chips, it could not afford to get such standards wrong.

Lechner asked how AMD would handle standards problems as more companies demanded the manufacturing of custom chips. Dowling said there would continue to be a large market for "standard components." But he acknowledged there would also be a growing market for custom and semi-custom parts. He said as companies like AMD continued to automate the design process for standard components, that would actually make it easier to produce more complex custom and semi-custom chips in the future. He noted that many systems manufacturers, such as IBM, were already designing their own chips using available tools.

Cheifet brought the discussion to a close by asking Dowling to predict where the microchip industry would be in five years. Dowling said he expected CMOS to emerge as the dominant technology as it could build chips of a very high complexity without a power consumption problem. Beyond that, he expected to see a continuing evolution of memory technology towards 4-megabit microchips and beyond, especially as the per-bit prices for producing those chips continued to decline.

## AMD's Bit-Slicing Strategy to Enter Microprocessor Market

Unlike many of the companies featured on *Chronicles* to this point, AMD is very much still in business. And the Am2901 integrated circuits had a surprisingly long lifespan. At a 2009 panel hosted by the Computer History Museum, former AMD executive Bob McConnell said the company continued to produce revisions of the 2901 until around 2000.

AMD itself was started in 1969 by Jerry Sanders and seven other individuals, all of whom had left Fairchild Semiconductor. Sven Simonsen, one of those co-founders, said at the Computer History Museum panel that AMD only raised about $1.5 million at startup. Originally, the new company served as a second source for the Fairchild 9300 integrated circuit. But after Intel introduced its early line of microprocessors, such as the 4-bit Intel 4004, an AMD designer, Clive Ghest, wrote a paper pushing the company to make its own microprocessor.

Ghest told the panel that his initial thought was simply to second source the 8-bit Intel 8080, because he didn't think AMD could afford to make its own competing MOS microprocessor. But then he realized AMD had a lot of expertise with bipolar technology, which would allow them to build their own microprocessor.

As Phil Downing alluded to in his *Chronicles* interview, bipolar was faster than MOS. But you could not pack as many components onto a bipolar microchip. That led AMD to rely on "bit-slicing" for the Am2901. Ghest explained:

> [I]t was inevitable if you were going to build a high-speed bipolar microprocessor in those days that it had to be several chips. And one of the elements which was, I think, the most interesting from the customers' viewpoint, was the arithmetic processing unit. Normally in computers in those days, the word length for the computer was anything from like, 16 to 32 bits. And you certainly couldn't put that much on one chip. So what you could do is to slice that unit into several pieces. And therefore one of the most logical ways of doing it was to build a four-bit slice. Then you could use several of them to build up an 8-bit or 12-bit, 16, or 32-bit microprocessor. So we decided on a 4-bit high-speed bipolar processing unit. And that became the 2901.

The other microprocessor discussed by Downing on *Chronicles*, the Am29116, was released by AMD in 1981. It was a 16-bit bit-sliced processor. Bob McConnell recalled in 2009 that the 29116 was "slow to take off" in the market. Yet according to The CPU Shack Museum, the Am29116 was manufactured well into the late 1990s and widely used "in control systems, video systems, telecom products and many military applications."

As for Phil Downing, he began working at AMD in the mid-1970s. Like AMD's founders, Downing also had a prior stint at Fairchild. He stayed at AMD until 1989. He then joined Zilog, Inc., the manufacturer of the Z80 microprocessor, where he was named senior vice president for technology in 1992 and later worked for National Semiconductor.

## Notes from the Random Access File

- This episode was recorded on October 19, 1983. It is available on the Internet Archive, which lists a publication date of April 23, 1984.
- If, like me, you don't know anything about arithmetic and logic units (ALUs), which Stewart Cheifet asked about during the discussion of the Am2901C, I recommend the Crash Course Computer Science video on the subject presented by Carrie Anne Philbin of the Raspberry Pi Foundation.
- Gary Kildall did not appear in this episode. But he was apparently quite familiar with the Am2901. At the Computer History Museum panel, Former AMD engineer John R. Mick said the company produced a development computer for its microprocessor called System 29, which signed the "first OEM contract" with Kildall to use Digital Research's CP/M operating system.

# Chapter 12: The Xerox Color Laser Printer

This episode contained what Stewart Cheifet would later describe as one of the classic "near disasters" involving a product demonstration on *The Computer Chronicles*. The subject was the first Xerox Color Laser Printer, which was actually a prototype not yet available for sale when this *Chronicles* episode taped in October 1983. Cheifet recounted the event to Tonya Hall of ZDNet in a November 2020 interview:

> We introduced the very first color laser printer on the show by Xerox. I figured it'd be a printer–it was the size of two Volkswagens! We had to take the doors off the studio to get this thing inside. And it took three engineers three hours to get this thing hooked up. And I said, 'Can we test it now?' 'Yeah, let's test it.' We went to the print button, hit the print button, and smoke came out of this printer. No output–smoke. I said, 'Look, we've got to get this thing to work, guys.' Finally, another hour later, they pressed the button and I saw the most gorgeous color output I've ever seen in my life from a computer.

Smoking printers notwithstanding, this episode was essentially a product demonstration showcase for Xerox and its Diablo Systems division, the latter of which was marketing commercial printers in late 1983 using a variety of technologies.

Herbert Lechner was back in the co-host's chair this week. Cheifet opened by asking Lechner why it seemed that printer technology had lagged behind computer technology in general. Cheifet noted the printer was the slowest component of the system, the prices had not come down that much, and it was often the first part of a system to break. Why was that? Lechner said it came down to the fact that the mechanical components of printers had not seen the types of breakthroughs that we'd seen in the electronic areas. Manufacturers were well aware of this problem and were trying to become less dependent on the purely mechanical functioning of printers by developing more electronic-controlled components.

Cheifet then narrated our B-roll for the week, which briefly reviewed printer technology. He started with the census-taking machine invented by Herman Hollerith in 1890, which only had one task: to collect and tabulate large numbers more efficiently than humans. Like other early mechanical computers, the Hollerith machine only had limited memory and the printout of the stored information was limited to the final computation.

As computers became less mechanical and more electronic, Cheifet said, information retrieval also became more flexible. Once data could be magnetically encoded onto a reel of tape, that data could now be changed or erased instantly and the recorded medium used over again. This made printed information–i.e., hardcopy–unnecessary as long as the data was used only within the functions of the machine. Information only became more perishable when video display terminals became standard computer components.

But while today's computers may have reduced the need to stockpile written information, Cheifet noted that a quick look at the typical modern office revealed that one of the most common peripheral devices was now a printer. Popular *impact printers* like the daisy wheel still operated much like a typewriter, albeit with fewer moving parts and greater speed. And while advances in printer technology to date had not matched the progress on the electronic side, the trend away from mechanical complexity had brought paper-and-ink printing to a new level of sophistication. The focus was now on developing non-impact printers using heat, electricity, and photo-type setting–and even lasers that burned characters into the paper.

## "How Fast Are the Squirts?"

Cheifet and Lechner introduced two guests from Diablo Systems, Steve Rea and Paul Shapiro. Lechner asked for an overview of the types of printers currently available for computer users. Shapiro explained there were two basic categories of printer: impact and non-impact. And within each of those categories there were different technologies available. For example, dot matrix printers were a type of impact technology, while inkjet printers were considered a non-impact technology. Within these groups, Shapiro noted there were also various cost-performance ranges, in addition to the differing qualities of the print, which were measured in spot density.

Cheifet asked how a user would know what printer was right for him? Rea said impact printing was good if you wanted to produce a hard copy that was good enough for letters and mailers. But if you needed high-speed output you might want to go with thermal printing technology, which involved a thermal transfer to plain paper. And if you needed color output, you had to go with an inkjet.

Cheifet followed up, asking how a consumer would chose between two printers using similar technologies, such as dot matrix and daisy wheel. Shapiro said again, the answer came down to the quality of output you were looking for as well as the speed. If you needed letter-quality, fully formed characters, he would gravitate towards the daisy wheel. But if you only needed “draft” quality and wanted to focus on speed, dot matrix might be the way to go.

Lechner asked where thermal printers fit in. Rea said Diablo’s thermal printer was very high speed and could output about 6 pages per minute. In contrast, a daisy wheel only produced 1 or 2 pages per minute. The inkjet was even slower because it worked with color.

Rea then conducted an EXTREMELY NOISY demo of Diablo’s daisy wheel printer. He note this technology had been on the market for many years, having been introduced by Diablo in the early 1970s. He explained the printer relied on a character wheel with 192 fully formed characters. This allowed for printing scientific notation and other specific applications. He noted that with a single print wheel the printer could print 33 different western European languages. Shapiro added the current trend with daisy wheel technology was to add even more characters per print wheel. New accessories were also popular, such as the double-bin feeder used on the demo model. This made it possible to increase the daisy wheel printer’s output over time. In response to a follow up question, Rea said the running speed of the daisy wheel was about 40 characters per second.

The group then moved to a thermal printer. Rea noted this was a much quieter unit compared to the daisy wheel. It also cost more because you were getting extra speed from the thermal printer. Shapiro added the thermal unit produced about 6 pages per minute–as opposed to roughly 2 from the daisy wheel–yet it was only twice the price. In terms of spots-per-inch, the daisy wheel was also more dense (450 SPI) than the thermal (200 SPI).

Lechner then asked about the inkjet printer–specifically whether this was where the market was moving. Shapiro said as graphics became a more important part of personal computer use, the market was gravitating towards graphics and color, especially with machines like the Apple Lisa that had bitmap graphics–i.e., the ability to control each and every bit on the picture tube and every bit on a page.

Lechner asked about the interfaces used to connect printers to computers. How would a customer know they were buying something where the plug would fit? Rea said Diablo relied on all-purposes interfaces. he agreed the typical PC user wanted “plug and play.” As such, Diablo printers could run using a standard parallel or serial connection.

Rea then demonstrated the final of the three Diablo printers, the Series C color inkjet. He explained the Series C utilized “drop on demand” technology, which enabled the printer to squirt a drop of ink per nozzle onto the paper. Lechner asked, “How fast were the squirts?” Rea said it amounted to 20 characters per second for text. The printer’s four ink colors could also be mixed to produce a total of 7 different colors (and 8 if you included white). He added the inkjet was very quiet and could be placed next to an individual workstation. Cheifet asked about the cost. Shapiro said the Series C was slightly less expensive than a fully formed character printer and about one-fifth the cost of a thermal unit.

Cheifet ended the segment by asking about the limits to thermal printing, such as the need for special paper and the inability to use fan-folded paper. Shapiro said that was only partly correct. Diablo relied on thermal transfer rather than “direct” thermal transfer. Thermal transfer allowed for the use of traditional bond paper. As for fan folding, it wasn’t necessary as you could use pre-cut sheets.

## “I Will Spray Some Freon”

The final segment focused on the previously mentioned Xerox Color Laser Printer. The segment began on the set with Lechner talking with Dr. John Urbach, the manager of optical science for Xerox PARC. Urbach provided a broad overview of how laser printing technology worked. He noted it was similar to that of a [CRT] television set. Only in this case a finely focused beam of light from a laser was scanned across a line. It was turned on and off at a very high rate using a modulator, and this painted out regions of illumination (or lack thereof) along the scan line. This scan line could then be repeated over and over again, and some kind of light-sensitive recording medium was then moved along underneath the scan line to generate a 2D image. This was similar to an inkjet, but the laser printer was faster and produced a higher resolution.

Lechner asked just how fast the laser printer was. Urbach said in terms of pages, it varied between 3 and 10 per minute. Lechner then asked for more detail about the “flexibility” offered by a laser printer. Urbach said that since you could paint out every possible resolution on the page just like a television, you had the ability to create a completely arbitrary map of any type of image information that a computer could generate. That said, Urbach added that most computers were not presently flexible enough to take full advantage of this capability. But if a computer could generate an image, then within some very broad limits the laser printer could print it.

The focus then shifted to Stewart Cheifet and Xerox research engineer Douglas N. Curry, who began the demo of the actual color laser printer–which Urbach emphasized was still an experimental device and not a standard product. Curry briefly explained that he would print a color image of a baboon. To do this, you had to superimpose three separate images of magenta, yellow, and cyan. The three layers were first printed separately before the final image.

There was then a very awkward cut, which I’m guessing is where they had the issue with the smoke. Cheifet was now seen with another Xerox employee, Tibor Fisli, who supervised the final printing of the baboon image (which is seen in the thumbnail for this blog post). Fisli also briefly reviewed the laser printing process as previously described by Urbach.

After reviewing what Cheifet called the “impressive” final printout of the baboon, Fisli showed a duplicate of the actual laser printing assembly from the Xerox Color Laser Printer. Cheifet asked if it was possible to see the actual laser beam. Fisli, showing a complete disregard for the Ozone, said, “I will spray some Freon,” which then made the path of the laser visible.

Back on the main set, Lechner asked Urbach about the future. Would we see more laser printers? Would they come down in price? And would it displace the other printer technologies discussed today? Urbach said the answer to all of those questions was “yes”. There would be more laser printers and they would come down in price. Early laser printers were fast but expensive, as they cost hundreds of thousands of dollars. The price had now started to come down rapidly thanks to the use of junction diode lasers, which would replace the larger helium neon laser used in the earlier demonstration unit.

## Diablo’s Short Run as Daisy Wheel Queen

I don’t have much to say about the people or products in today’s episode. Tibor Fisli apparently passed away in 2001. The only other person I came across information for was Steve Rea, who bounced around a number of well-known technologies but seems to have spent the most time at Apple, with stints in Cupertino from 1988 to 1997 and again from 2009 until January 2021.

Diablo Systems was founded in 1969 and manufactured early disk drive systems and daisy wheel printers. Xerox acquired Diablo in 1972 in an all-stock deal valued at around $28.6 million, according to *The New York Times*. As a Xerox division, Diablo’s best known product was the Diablo 630, the daisy wheel printer that was likely used in the *Chronicles* demo.

Xerox, of course, is very much still in business and continues to manufacture laser printers.

# Notes from the Random Access File

- This episode is available at the Internet Archive, where it is dated May 14, 1984. The actual recording date was October 10, 1983.
- The Hollerith machine mentioned by Stewart Cheifet during the B-roll was apparently used by the Census Bureau well into the 1950s, when it was replaced by electronic computers.
- Although the color unit seen in this episode was still a prototype, Xerox did bring the first black-and-white laser printer to the market back in 1977, the Xerox 9700.
- According to an article by Erik Arctander and John Free for the February 1984 issue of *Popular Science*, the Diablo Series C inkjet was actually made by Sharp and retailed for $1,200, which was considered a "surprisingly low cost" at the time.
- In comparison, the Diablo thermal-transfer printer would set you back about $5,000, according to the April 4, 1983, issue of *ComputerWorld*.
- If you want to learn more about the daisy wheel technology, there's a fun video originally produced by Xerox and later posted to YouTube by the Computer History Museum that demonstrates the process of making the actual daisy wheels.
- The casual spraying of Freon seen in this episode would effectively be banned by international treaty a few years later. The Montreal Protocol on Substances that Deplete the Ozone Layer was signed in 1987 by the United States and 45 other countries. It phased out the use of chlorofluorocarbons (CFCs), which included Freon.

# Chapter 13: The IBM PROFS

The episode I'm covering today was taped on January 18, 1984, four days before Super Bowl XVIII. That game would go down in computing history for the famous Apple "1984" commercial that announced the launch of the original Macintosh (later known as the Macintosh 128K). As this *Chronicles* episode aired the week after the Super Bowl, Stewart Cheifet devoted a good portion of the post-show "Random Access" segment to the new machine and what it might mean for Apple for the rest of 1984.

Before we get to that, however, the actual subject of this episode was *business applications*. In some sense, this was a follow-up to the earlier episode on integrated software that featured the doomed Apple Lisa. Only this time, the focus was on market leader IBM.

## The Promise of Paper-Free Offices

Once again, Herbert Lechner is back to co-host. The program opened with Cheifet demonstrating *Lotus 1-2-3*, one of the more popular business application programs available. Cheifet noted the "1-2-3" stood for the three basic business functions included in the software: database management, spreadsheet analysis, and graphics. Cheifet showed how to transfer a file out of a database and place it into a spreadsheet, then take that data and make a pie graph with just 1 or 2 keystrokes.

Cheifet noted that lots of businesses were now making the transition to the use of computers to solve their problems. In some cases that worked out well. But in other cases, people were not as happy with the process. So what made the difference between succeeding or failing in using computers for business? Lechner quipped that although the personal computer had been named Time's "Man of the Year" for 1982, he had yet to see any board of directors appoint a computer as presidents of any companies. More seriously, Lechner said the difference between success and failure in using computers, as with any business tool, laid in the hands and the skill of the person using it.

This brought us into some lovely office B-roll footage, where Cheifet explained how integrating computers into the workplace had evolved from the basic need and curse of every office–reducing, manipulating, and storing written information. Much like computers were originally used in science and math, today's business automation systems were paper and time crunchers that could squeeze ledgers and files onto disks, opening up flexible communications lines of enormous capacity.

And while there was no shortage of predictions on the impact of automation on the traditional workplace, Cheifet said one thing was certain–offices were being redesigned to accommodate this new level of information gathering and distribution. Glossy magazines were already promising "paper-free offices," where personal computers were ubiquitous. And just as business software becomes more integrated, hardware was also becoming a more integral part of desks, counters, and even briefcases, in the process absorbing more and more of the traditional desktop tools.

Cheifet noted that in some of the latest designs, communication and data manipulation were now combined into a single software module, bringing together telephone, word processing, spreadsheets, files, and electronic mail. Some systems now even had the ability to store digital voice messages!

Ultimately, Cheifet said, computers were suited for more than just front office use, but the computerized desk was still an example of automation applied to streamline the most time-consuming tasks. He noted that some estimates found that paperwork and phone calls alone made up 75 percent of all office activities.

## "Make Friends with the Future"

Back in the studio, Chuck Kinch and Stewart Reges joined Cheifet and Lechner. Kinch was vice president for products at ComputerLand, then a major computer retailer. Lechner asked Kinch about the most popular business applications sold right now. Kinch said the tendency was moving towards “horizontal” software packages as opposed to ’vertical” ones, with an emphasis on integrated packages like Lotus or SuperCalc 3 (a popular spreadsheet for CP/M systems). Lechner asked for clarification on what “horizontal” meant. Kinch replied it referred to software that could be used by a wide variety of business people, such as a word processor. A vertical application, in contrast, was something that would only be purchased and used by a particular group, such as an architecture program.

Lechner asked Kinch about the most common types of business customers he saw at ComputerLand. Kinch said it covered both ends of the spectrum: Fortune 1000 companies and small businesses.

Cheifet then turned to Reges, a computer science professor at Stanford University, and asked if there was a certain kind of business that was better off doing things the “old fashioned” way and not bothering with computers. Reges said people had gotten a little bit too much onto the “computer bandwagon.” These days, he snarked, if something was wrong the solution seemed to be “buy computers.” That said, over the next 5 to 10 years he believed that virtually every business would seek to incorporate computers in some way into their operations. Computers were seen as the “communications tool of the future,” and not having one would be like trying to run a business today without telephones.

Lechner pushed back a bit, noting that some people were not “delighted” with computer business applications. What made the difference between a happy and an unhappy customer? Kinch replied that if the customer was poorly organized to begin with and their systems weren’t working at all, a computer was not going to help. That’s why his approach was to make sure the customer did some preparation to get their systems in order before introducing automation. In that sense, the computer wasn’t necessarily the answer, just part of it.

Cheifet asked how a retailer like ComputerLand helped to “mesh” the variety of business applications available with the problems a particular small business customer might have. Kinch said the sales cycle largely consisted of “information gathering” by the salesperson, followed by analysis and educating the customer. Ultimately, the customer had to make the final decision "with a little guidance" from the salesperson.

Lechner asked about the type of educational program that was necessary. Reges chimed in, noting that in his administrative post at Stanford, he was in charge of the computer training program for the secretaries in his department. That meant screening applicants from the beginning and talking to them about computers. He noted there were a lot of people who were "afraid" of computers, which posed a major challenge to office automation. Indeed, Reges said that even managers–perhaps, especially the managers–had that fear. Lechner asked Kinch if he observed similar "fears" from his customers. Kinch said absolutely, and that's why ComputerLand tried to portray itself as a place where new users could learn about computers–after all, the company's slogan was "Make Friends with the Future."

Cheifet then asked Kinch to explain the basic types of business applications currently available. Kinch said it basically broke down into word processors, database management, and decision support systems (also known as spreadsheets). In particular, there had been an increase in demand for word processing and graphics packages.

Reges concurred that graphics were becoming more important, especially those programs that could display information from a database. He noted that at Stanford there was an emphasis on trying to eliminate duplication of effort. For example, he said a secretary might write a letter on a standalone word processor and send it to another secretary, who would enter the information onto an IBM Personal Computer. This information is later used by another person to generate figures, which then must be given to a "graphics wizard" to make a pie chart. This entire process would be much easier, Reges said, with an integrated software package.

Lechner asked if small businesses were actually now relying on computers to handle financial matters like their chart of accounts, general ledgers, payroll, tax information, and so forth. Kinch said that although this type of financial software was generally considered a vertical package, it was quite popular with business customers. Reges asked about electronic mail packages, something he noted would probably not appeal to small businesses. Kinch said that was considered part of a communications package, such as *PC-Talk* and *Cross Talk*, which had seen a significant increase in sales.

## “Levi’s Is People”

The final segment contained our product demo for the episode. Dan Carroll of IBM and Bill Blackmer of Levi Strauss joined Cheifet and Lechner to talk about the IBM PRofessional OFfice System (PROFS). Levi Strauss was a PROFS customer and Blackmer served as manager of the company’s productivity systems.

Blackmer explained Levi’s “three-tiered approach” to introducing office automation. The first tier was people. He noted the company’s slogan was “Levi’s Is People,” and that automation was a scary term to a lot of those people. It was therefore important to get everyone, especially the secretaries, involved in the automation process from the early stages.

The second tier was integration. Levi’s wanted to create an environment where secretaries and middle managers in Dallas could talk to New York, London, or anywhere else. This required an integrated system that could work off of Levi’s communications network and mainframe.

The third tier was planning. Blackmer said office automation started with the idea that the secretary was the main resource. But you also had to deal with middle managers and senior executives to get the information flow out of your corporate database to make decisions.

Lechner asked IBM’s Carroll how the company supported a customer like Levi’s? Carroll said first, IBM tried to take advantage of the customer’s existing networks and databases. Second, they looked at the specific needs for the six different types of workers that were typically found in the white-collar workforce: (1) managers and business professionals, (2) data processing professionals, (3) engineering and scientific professionals, (4) office or text workers, (5) commercial workers (i.e., retail clerks), (6) production and factory workers who operate machinery that collect information.

Blackmer then conducted the demonstration of PROFS, which was customized by IBM specifically for Levi's. He explained the system tied into a Levi's mainframe in Florida and provided the company's internal electronic mail and scheduling system. He used the system to schedule a hypothetical meeting with a co-worker. PROFS searched both employees' schedules and tried to find an an acceptable meeting time. Once a meeting was scheduled, PROFS would send an electronic message to both employees' terminals.

Cheifet asked Carroll if this was a typical use of PROFS. Carroll said yes–in fact, IBM used the same system internally. It was part of an iterative process of postulating a requirement for productivity, trying a solution, seeing if it works, and if it does replicating and expanding it throughout an organization.

Cheifet noted that PROFS had to cost a lot of money. Was it really worth it to save the trouble of calling a co-worker to schedule a meeting? Blackmer gave somewhat of a rambling answer, but he said that in the final analysis, automation was about make it easier to do "business work" rather than "office work."

## Jobs Bets the Farm on "Mini-Lisa"

Stewart Cheifet presented this edition of "Random Access," which was taped sometime during the week following January 22, 1984:

- Cheifet opened: "Macintosh–what else can one say?" People were either talking about the Macintosh itself or the Super Bowl commercial. Early reports from computer dealers indicated there was "tremendous consumer response" to Apple's new computer.
- At the same time, dealers were also concerned by Apple's plan to sell the Macintosh to universities at a 60 percent discount, as students might buy the machines and re-sell them for a profit, still undercutting the retail price by as much as half.
- Cheifet played a clip from Apple Chairman Steve Jobs' Macintosh press conference. Jobs said that while the Lisa had pioneered amazing software technology, with the Macintosh Apple had taken that

software and pulled it down to a price range that was affordable for individuals, not just businesses.

- Cheifet said that the Macintosh was “basically a mini-Lisa” with a 32-bit microprocessor, 1 built-in micro-floppy drive, a mouse, and a 9-inch black-and-white monitor. The retail price was $2,500. And while most reviewers had given the Macintosh “excellent marks,” some critics said the lack of a color display would limit the computer’s appeal.
- The accounting firm Peat Marwick announced that it would purchase 2,500 Macintoshes, to be carried by its auditors for on-site audits. Cheifet noted this would make Peat Marwick the first accounting firms to send its auditors into the field with computers.
- The success of the Macintosh was critical to Apple’s future, especially given the company’s poor first-quarter earnings report, which showed a 75 percent decrease in net profit, which was attributed to slow Lisa sales, price cuts for the Apple IIe, and heavy research and development costs related to the Macintosh.
- Meanwhile, Digital Electronics Corporation (DEC) reported a 40 percent increase in quarterly earnings, while IBM was up 25 percent.
- According to a “market research report,” IBM and IBM clones now accounted for 60 percent of the office personal computer market; that figure projected IBM’s market share would grow to 70 percent by 1989.
- IBM also announced it planned to begin “volume shipments” of the IBM PCjr, although supplies were still limited due to parts shortages.
- AT&T announced a joint venture with Gary Kildall’s Digital Research to distribute software based on AT&T’s UNIX System V operating system.
- Hewlett-Packard announced it was dropping plans to develop its own 32-bit computer and would instead focus on “a new breakthrough product” to release in 1985.
- Trilogy Systems announced a delay in shipments of its new supercomputer, reportedly due to problems with the metal coating of its prototype chips.
- Fairchild Semiconductor was taken off the list of approved Department of Defense suppliers after the Pentagon found the company failed to comply with government quality control standards.

- MicroPro, the company that produced the popular word processing program WordStar, announced its initial public offering for March 1984.
- A new report from a software industry group–presumably the Software Publishers Association–claimed that software piracy in the United States was costing the industry $20 billion per year. Cheifet noted some estimates found that as much as 80 percent of all software in use was pirated.
- Finally, Cheifet noted that Apple's $500,000 ad for the Macintosh had "caused quite a sensation." Some people thought it was "absurd," while others said it was "fantastic."

## One of the Earliest Corporate Electronic Mail Systems

The featured product in this episode, the IBM PROFS, started out as an in-house system developed in the early 1970s at IBM's laboratory in Poughkeepsie, New York. IBM introduced PROFS as a customer product in 1981. IBM's own history described PROFS as follows:

> It was built on top of IBM's virtual machine (VM) mainframe operating system, and ran on the mainframe's user interface, CMS (Conversational Monitor System). PROFS allowed users to send and receive notes and messages (a precursor of both email and instant messaging); maintain personal calendars; get automatic reminders of important events; schedule meetings and conference rooms; and store and retrieve notes and documents.
>
> The system included access to newsfeeds from thousands of internal and external sources, and also sported a full-text searchable product catalog so sales staff had instant access to IBM's vast product portfolio. And since the PROFS system was almost completely menu- and function-key driven, it was relatively easy to learn in comparison to command-driven systems.

In 1989, IBM renamed and replaced PROFS with OfficeVision/VM. By the mid-1990s, the Internet effectively rendered the system obsolete. However, IBM continued to support OfficeVision/VM until 2003.

## Notes from the Random Access File

- You can watch this episode at the Internet Archive. This episode also featured a new director, Peter Nichols.
- ComputerLand was at the height of its success in 1984, with several hundred franchise stores throughout the country. Ten years later, the company's owners changed the name to Vanstar and shifted to "direct sales and contract services to large companies," according to a United Press International report from March 1994. Vanstar ceased operations around 1999, although the ComputerLand name continues to be used today by some former franchises.
- Chuck Kinch remained with ComputerLand until 1988. He later spent four years as president and chief operating officer of New Horizons Computer Learning Centers in the late 1990s, and seven years as CEO of New Horizons Learning Centers for China & Singapore in the 2000s. Today he's an independent business and management consultant based in Dallas.
- According to Stuart Reges, Stanford University fired him in 1991 for "protesting the war on drugs." Reges then turned his attention to politics, serving as the Libertarian Party's national director for a time. He returned to academia in 1996, first at the University of Arizona and later the University of Washington.
- Dan Carroll recently posted a comment on the YouTube video of this episode: "I was a fairly young IBM guy participating in this episode. Looking back, I would have climbed up on the desk to shout 'this stuff will change the world.'"
- The Digital Research-AT&T joint venture never went anywhere. In April 1985, AT&T executive James Edwards told InfoWorld the original goal was to port "as many DOS software packages as possible" to UNIX, but "that became less interesting as our strategy changed."
- Trilogy Systems also never managed to ship that supercomputer. In fact, Trilogy was one of the biggest failures in pre-Internet tech history, losing nearly $230 million in investor funds, according to a 1986 *Newsweek* article.

- I’m not certain, but the Hewlett-Packard “breakthrough product” that Stewart Cheifet alluded to might have been the HP 110 portable computer.
- Steve Jobs in a suit and tie is just creepy, man.

# Chapter 14: Express EasyScan and The Sales Edge

In a bit of eerie foreshadowing, this episode of *The Computer Chronicles* from January 1984 opened with a discussion of a global pandemic. Not a real one, of course, but a computer simulation. Stewart Cheifet and Gary Kildall played with a game called *Epidemic*, released by Strategic Simulations. Cheifet explained this was an example of how someone could use a computer to choose among alternatives, make a decision, and establish a complex strategy for solving a problem. In this case, the problem was a global pandemic caused by a virus brought from outer space by meteors. *Epidemic* provided the player with information about the spread of the virus and offered alternatives, such as developing a vaccine, for containing the pandemic.

The actual topic of the episode was simulation games, but rather computer-aided decision making in business. Also known as *decision support systems*. Kildall said this was really a catch-all phrase for a lot of different concepts. But the basic premise was that the computer was helping a manager or decision-maker with long-term strategies and providing an "able assistant." The underlying technology could involve the use of expert systems, relational databases, or even spreadsheets.

Cheifet then provided his customary narration over some B-roll footage. He reiterated that decision support systems encompassed a broad range of analytical tools, from forecasting programs that calculated trends and predicted outcomes to expert systems that could draw its own conclusions and make decisions. Whether or not these decision aides required some form of artificial intelligence, they were already in routine use, having proven their efficiency and accuracy in a number of applications.

For example, Cheifet said a "major American railroad" currently used an in-house decision support system to assist in long-term financial and operations planning. Based on forecasts of future demand, the system reviewed records of train capacities, labor costs, taxes, fuel costs, and revenues to predict the financial impact of alternative strategies.

Cheifet pointed to another early decision support system called *Prospector*, which analyzed the mineral content of geological formations to recommend drilling sites. *Prospector* had many features associated with artificial intelligence. That is to say, it took user inputs regarding various rock types and used that information to draw conclusions and make independent recommendations. However, Cheifet noted that while *Prospector* relied on "probabilities" to draw its conclusions, the more common types of decision support systems in use today required "known facts." But future research was likely to narrow the gap between programs that laid the groundwork for decisions and those that could make them.

## Expert Systems vs. A Well-Trained Assistant

Back in the studio, Mike Thoma and Steve Weyl joined Cheifet and Kildall. Weyl was a co-founder and vice president at Syntelligence. Kildall asked him to provide a commercial example of a decision support system. Weyl said his company specialized in what were known as "knowledgebase" or "expert systems," which tried to bring the actual judgment of an expert to the desktop of a manager or business professional who needed to make a particular decision. For example, Syntellience was now working on a system to help commercial lending officers at banks decide whether to make loans to small companies.

Cheifet then asked Thoma, a manager with Management Decision Systems, if a decision support system had to be an "expert" one like what Weyl described. Thoma replied no. He joked that when he asked his 78-year-old father, who worked as a consultant, if he needed a decision support system, his father replied no, he made the decisions. What the father wanted was software that could provide an "alternative evaluation" of a situation. Thoma said this boiled down to providing a "very well trained assistant" who would carry out an analysis at the user's direction.

Cheifet asked about the importance of interactivity to a decision support system–did it have to respond uniquely to the questioner? Thoma said he believed that interactivity was important. The decision-making process was one of innovation, he said, meaning that you had an idea and needed feedback. The system's responsiveness, in turn, would help generate additional ideas.

Kildall noted these decision support systems required a connection to a larger computer system with database access to provide the basic facts and figures. Thoma agreed that some sort of computer connection was required, although the size depended on the problem. Cheifet asked Weyl if Syntelligence's system depended on a mainframe. Weyl said yes, the systems they had initially developed required a mainframe, but given the present direction of personal computers, he expected that within a few years "at most" their systems would run on future PCs.

With respect to interactivity, Weyl added that Syntelligence had spent a lot of time ensuring that its systems provided full explanations of how they reached their conclusions. He said managers could not simply look at the computer and say it "told me to make that loan." The computer had to work like a staff assistant, and like a person it needed to give a full explanation of its reasoning.

Cheifet then asked Thoma to demonstrate his company's product, *Express EasyScan*. Thoma created a hypothetical scenario where the user was looking to buy a software company. The software initially identified 62 possible companies from a database provided by Standard & Poors. Initially, the 62 companies are listed in alphabetical order by their stock ticker. Thoma then entered parameters to narrow down the field: Specifically, he queried for just those software companies that had reported a sales growth rate of 30 percent over the five-year period up to 1982. This brought the list down from 62 to 12. Thoma then ranked those 12 using an additional criteria, profit margin. Cheifet then asked for a bar graph comparing just the top-two companies by profit, which Thomas produced.

## Taking a Personality Quiz to Improve Your Sales Performance

Jim Chapman joined Cheifet, Kildall, and Weyl for the last segment. Chapman was a vice president at Human Edge Software. His company produced *The Sales Edge*, which unlike *Express EasyScan* was designed to work without a mainframe on a standard IBM Personal Computer. Chapman explained the software provided a step-by-step strategy report for sales, negotiation, and management.

The salesperson would start by answering 81 pre-loaded questions about themselves, essentially generating a snapshot of their characteristics and attitudes towards sales. Kildall semi-seriously answered a few of the questions as a demonstration. The questions were actually statements that the salesperson would either "agree" or "disagree" with, such as:

1. Sometimes a sales call makes me nervous.
2. I like the freedom to travel that a sales job gives.
3. I enjoy swapping stories with customers.
4. I am sensitive to criticism.
5. I enjoy putting one over on a customer.
6. I like talking in company meetings.

Chapman explained the final answers were saved to a diskette. He then showed how the salesperson could call up a previously completed customer profile, in this case for a hypothetical customer named Monroe. Monroe's profile displayed a number of adjectives describing him, such as that he was "not talkative" and "apprehensive." Chapman said this information was then used to generate a more comprehensive report about the customer. Cheifet clarified that this was used to determine what strategy the salesperson should take in dealing with the customer. Chapman said that was exactly right. In particular, when a salesperson met the customer for a second time, they would already have a feel for what they were like, what their strengths and weaknesses were, and so forth.

Kildall asked how successful the software was in an actual sales situation. Chapman claimed it was very successful. He said he used *The Sales Edge* himself and so had approximately 1,000 customers around the country during "beta testing." And the feedback was that the program was useful.

Cheifet pushed back a bit, asking if this approach was not "superficial," and if you really could make a business decision based on a bunch of quick "yes or no" questions. Chapman insisted you could. He noted this was actually a cross-fertilization of two areas of research: personality and sales. He said the software helped a salesperson select the most appropriate strategy based on how a customer thinks.

Kildall reiterated the importance of this type of software telling the user why it made a certain recommendation. He said it would be hard for him to trust a program on a disk that told him what way his business should go. Chapman pointed out that *The Sales Edge* actually required *multiple* diskettes. For example, a separate disk was necessary to generate the final customer report, which Chapman demonstrated.

Cheifet asked Weyl for this thoughts on how far he thought decision support systems could go. Would we see programs like this used in making legal or governmental decisions? Weyl said he thought that was still 3 to 5 years away. The types of events that would influence the head of a government were quite a bit more difficult to model. Kildall noted there was a danger that people could become too reliant on decision support systems in the future. Weyl pointed out that was a danger with computer systems in general. Ultimately, he believed these systems would help improve decision-making, but they couldn't be followed slavishly.

## Commodore Profitable Despite Losing Executives

This edition of "Random Access," recorded in late January 1984, was the first one reviewed here not hosted by Stewart Cheifet. Susan Bimba, another KCSM-TV employee, served as the guest presenter.

- The federal government was investigating 12 semiconductor manufacturers suspected of selling faulty and inadequately tested chips. The focus of the investigation was National Semiconductor, whose officials believed criminal and civil charges were imminent. Bimba noted that in 1982, National Semi admitted its chips did not receive the "required amount of heat-stress testing" required by its government contracts.
- The California legislature was considering an anti-hacker bill that would make it a misdemeanor to break into any computer system even

if there was no malicious intent; if there was such intent, the charge would be bumped to a felony.

- Commodore reported a second-quarter profit of $50.1 million, more than doubling its figures from the previous year. Meanwhile, four Commodore executives resigned from the company just two weeks after CEO Jack Tramiel left for Atari, although Commodore claimed there was no connection between the two.
- Commodore also announced it would delay the release of its 264 computer, which was originally scheduled for April 1984.
- Following last week's introduction of the Macintosh, Apple President John Sculley reorganized the company, reducing the number of divisions from 6 to 3. The new divisions were (1) Apple II & Apple III computers; (2) 32-bit systems, i.e., the Macintosh and Lisa; and (3) accessory products, including printers, keyboards, and modems.
- Bimba said Apple also planned to open a new factory in Mexico to comply with that country's requirements that computer sellers operate local manufacturing plants. Apple already accounted for 40 percent of Mexico's personal computer market.
- Hewlett-Packard and Vitalink Communications announced plans to start a hi-speed satellite communications network that would allow personal computer users to access a network without a telephone.
- Storage Technology Corporation laid off 400 workers at its Santa Clara plant, giving employees just 2-hours notice. Bimba said the plant had been working on a large, IBM-compatible mainframe, but a decline in sales during 1983 led to the layoffs.
- Finally, Bimba said, if you "joined the chorus that's saying video games are a thing of the past, you may have to change your tune soon." Bimba noted computer games were now becoming more popular and that a number of Silicon Valley firms had switched from video games to computer games to take advantage of this new trend. That said, gaming enthusiasts said both could coexist.

## "Human Edge" Not Enough to Become Profitable

The most interesting thing to me about the two products demonstrated–Express EasyScan and Sales Edge–was the wide disparity in price. And I'm really underselling that point. As noted in the episode, Express EasyScan was mainframe software. And it carried a mainframe-level pricetag: $50,000. That was according to a February 1985 article in *ComputerWorld*, which said the financial consolidation and reporting system module–i.e., the S&P database of company information–added another $120,000.

In contrast, *The Sales Edge* was desktop software that you probably could have bought in any computer store. An April 1985 report in the *Chicago Tribune*, said *The Sales Edge* only cost $250. And it worked on the IBM PC and Apple computers, including the Apple IIe and the Macintosh.

Unfortunately, the *Tribune* article also noted that Human Edge Software itself was "awash in red ink," having posted a $1 million loss in 1984, although company founder James Johnson said he expected to be profitable in 1985. Johnson himself was a clinical psychologist by training, and he previously "developed computerized diagnostic systems used by psychologists and psychiatrists," before starting Human Edge. *The Sales Edge* was Johnson's first commercial product, which was followed by *Management Edge*, *Communications Edge*, *Negotiation Edge*, and *Mind Prober*, according to the *Tribune*.

My guess is that Human Edge never did attain profitability. It went out of business in 1988.

## Notes from the Random Access File

- This episode was recorded on January 24, 1984, and is available at the Internet Archive.
- Management Data Systems (MDS) only lasted until around 1986, when it merged into another company, Information Resources, Inc.
- Mike Thoma stayed with MDS until 1985 when he moved over to Oracle Corporation as its vice president of marketing for three years. He went on to hold a number of marketing positions in the tech industry, including eight years at Actuate Corporation from 1999 to 2007. In recent years, he's moved into executive coaching and career mentoring.
- Steve Weyl was a co-founder of Syntelligence and left the company after it went bankrupt in 1991. He worked at Apple for a couple years

after that and later served as President of Ricoh Innovations from 1996 to 2007. He’s currently a principal at Xeolux, LLC, a brand and intellectual property development company.

- From what limited information I could find, “Random Access” guest presenter Susan Bimba passed away in 2017.
- The Commodore 264 project mentioned by Bimba was eventually released as the Commodore Plus/4 in June 1984.
- National Semiconductor was indeed indicted by the U.S. Department of Justice in March 1984. The company agreed to resolve both criminal and civil charges in exchange for paying a $1.75 million fine.

# Chapter 15: Space Shuttle, Excalibur, Pinball Construction Set, and Dr. J vs. Larry Bird

Even if you're only a casual gamer, there are probably a few video game designers whose names you're familiar with, such as Sid Meier, Todd Howard, and Shigeru Miyamoto. From the early days of computer gaming, there was a concerted effort to promote certain "superstar" designers to help personalize and sell games to the public. This next episode of *The Computer Chronicles* featured three such designers from the early 1980s, as well as an executive whose name would become synonymous with computer and video game production in the decades that followed.

## From The Claw Machine to *Dragon's Lair*

Computer games were the overall subject of this episode. Stewart Cheifet and Gary Kildall opened by demonstrating the "granddaddy of computer games," Atari's *Pong*. Cheifet noted that games were an important part of launching the growth of the personal computer market. Kildall agreed it was "certainly a part of it," adding that many personal computers like the Apple II offered color bitmap graphics. This allowed game designers to do dynamic things that were not possible with the teletype and scrolling displays used on older computer systems.

Cheifet then narrated the obligatory B-roll for the episode, which briefly sketched the history of computer gaming from the penny arcades of the early 20th century, which offered "simpler, more mechanical amusements," such as the claw machine. Like today's computer games, a player needed manual dexterity in competing for points and rewards. But computer games brought a level of fantasy that no mechanical game could match. Instead of escaping from a school room or an office, Cheifet said a computer game allowed the player to "leave the earth" or move to a different universe. And modern computer games offered different kinds of challenges from fast action games to maze-type games and even wargames.

Cheifet noted that in one of the latest adventure games–which he didn't name but appeared to be *Dragon's Lair*–animated images were stored on laserdisc and recalled in sequence following a specific story, but also took into account the player's responses. If the player moved the stick in the right direction, they would be rewarded with a successful escape. But if the player made a mistake, the game would show an alternate video sequence.

## Just Like Flying a Real Space Shuttle

For the next segment, game designers Jessica Stevens and Chris Crawford joined Cheifet and Kildall. Kildall opened by asking them to clarify: Did they call themselves "designers" or "programmers"? Crawford said he called himself a designer since most of the "sweat and blood" occurred during the design phase and not the programming phase. He said he spent a great deal of time worrying about what his audience would experience in the game, and only towards the end of the project did he actually write code.

Kildall asked what elements made a game design interesting. Crawford said that depended on what he was trying to accomplish. He started by setting a goal, i.e., what effect he wanted the game to achieve. He wanted to somehow communicate a message to the audience. That message then dictated the topic and style of the game.

Cheifet observed that Crawford sounded more like a writer than a computer programmer. He asked Stevens if that's how she saw herself as well. Stevens agreed with that characterization. She said they were writing a type of novel, creating a vicarious universe within the computer program. They had to model everything they wanted and put every small detail into the game. Stevens noted that for her game, *Space Shuttle: A Journey into Space*, she put in a year-and-a-half of research to understand the subject before distilling the important aspects out and putting it onto the cartridge.

Stevens then provided a demonstration of *Space Shuttle* on an Atari 2600. The game opened with an automated demo of a space shuttle flight, which was based on the real second mission of the Space Shuttle *Columbia* in November 1981. The goal was to launch the space shuttle, connect to an orbiting space station (based on *Skylab*), and return to Earth for a landing at Edwards Air Force Base in California. Stevens also pointed to the various real-life details in the game, such as the "yellow flash" the astronauts see after jettisoning the shuttle's solid-rocket boosters.

Cheifet said it was incredible Stevens could do all this on an Atari 2600, which was not even a personal computer. How did she manage to squeeze the entire game to fit? Stevens said she had to model the universe and then fit everything piece-by-piece. She started with the basic mission and then added features and functions along the way. And in fact, she was not happy with her initial results–she still had a list of 146 separate items she wanted to include. So she spent three additional months adding them. The goal was to be as accurate and complete as possible. Kildall asked how big the final game was in terms of memory. Stevens said the game fit on an 8 kilobyte cartridge and took 13 months of programming.

Cheifet pointed out that one way Stevens adapted her game to the limited 2600 hardware was to make "functional" use of the console's switches, which were normally reserved for system-level uses like game reset or selecting television type. Stevens said all of these switches were controlled by the computer–the Atari–so she redefined them to perform game tasks, such as operating the space shuttle's engines or opening its cargo bay. All of these modifications made it necessary to include an overlay that fit over the console and served as a "cheat sheet" for the player. The game also had a 30-page instruction manual that provided the details on how to fly the space shuttle–basically, a small flight manual.

Cheifet noted that piloting a space shuttle was a difficult task. Could a child or a normal person actually pull this off in the game? Stevens agreed it would take an average person "years of work" to learn how to pilot the actual shuttle. But she also looked at how NASA trained its astronauts. They started with book exercises and watching film strips. Her game followed a similar approach. It began with watching a demo flight completely controlled by the console. This was essentially a film strip. Next, the player took a training flight, which compensated for user error and mimicked the simulators used by the real astronauts. The third flight was then the "real thing."

Kildall said this was basically an educational game. Stevens described it as "vicariously educational." The player wasn't simply sitting there looking at flashcards. They were learning about one of the most technologically advanced items ever developed by humans. Once the player mastered the game they had effectively learned about what an astronaut did on a space shuttle mission.

Cheifet then shifted back to Crawford and asked about his own game, *Excalibur*, which was available on Atari's 8-bit computer line. Crawford said the game was about "leadership." His motivation was the 19th century Prussian general Carl von Clausewitz's famous axiom, "war is a mere continuation of policy by other means." Crawford said he was tired of wargames that glorified war. He wanted to how that war was sometimes unavoidable but must never be entered into cavalierly. So he wanted *Excalibur* to teach concepts of leadership.

In the game, the player assumed the role of King Arthur, who must unify Britain. Crawford noted that unification was different than conquest–the player had to convince people of their authority. That involved more than brute force. Cheifet clarified *Excalibur* was designed for the Atari 800. Crawford said yes, his game required 48 kilobytes of memory and a disk drive. The game actually required swapping multiple disks, as the total game included 66 kilobytes of object code written in assembly language.

## Electronic Arts: Simple, Hot & Deep

The final segment focused on Electronic Arts. The company's founder and president, Trip Hawkins, appeared with Bill Budge, the designer of EA's *Pinball Construction Set* (PCS). Kildall opened by asking what made a game successful? Budge said it was hard, given that you couldn't just ask people what they wanted to see on the computer. The person writing the program had to have an "inner conviction" and write a program they wanted to use. Indeed, his goal when he started making PCS was just to get it to work. Kildall asked where he got the idea. Budge said it was a “funny, indirect process.” He was tired of simply playing games and he thought it might be fun to make a game where other people could write games. Kildall quipped this made PCS a “meta-game.”

Kildall followed up, asking if there was a special segment targeted by PCS. Budge replied it really was for the more “avant-garde” computer users. It took awhile for an audience used to games like Pac-Man to find PCS. His game did not register at first but the audience had started to grow to the point where they were now seeing good results.

Cheifet asked Hawkins what he looked for in a successful game from the business side. Hawkins said PCS was a good example of what he looked for, noting it was now one of the top-10 selling games according to *Billboard*. Overall, Hawkins said EA’s philosophy was to look for games that were “simple, hot, and deep,” which he broke down as follows:

- *Simple* meant the user did not have to read a lot of instructions; PCS allowed players to get started with pre-made pinball table or build one of their own.
- *Hot* meant the game should take full advantage of the computer’s sound and graphics capabilities.
- *Deep* meant letting the player “make their own game,” which helped to extend the life of the product and get people coming back to play again and again.

Budge then demonstrated PCS on an Apple II. The gameplay was based on using a joystick to move various pinball machine parts around to create a pinball machine. Budge joked that young children jut wanted to “grab a bunch of bumpers” and put them on the table, which serious pinball aficionados would scoff at.

Cheifet asked Budge about his next project or goal. Budge replied he wanted to extend the “construction set” idea further. The challenge would be designing the box of parts. With pinball that was relatively simple, as a pinball machine had a small box of parts. You didn’t need to think about more abstract concepts. But with a more general construction set, it was not even clear what the parts should be. Budge said it was almost like inventing a new language.

Cheifet asked Hawkins if computer games were here to stay, or if they would suffer the same downturn as arcade games, which seem to have peaked. Hawkins noted that computer games were fundamentally different video games, mainly because computer technology was more extendable. For example, all of EA's games came on floppy disks, each of which held more memory than a coin-operated arcade machine. Indeed, you could get a program 3 or 4 times the size of an arcade machine on a computer disk.

Kildall noted that computer games had already advanced from "relatively unsophisticated" offerings to something that required a lot of programming. Cheifet said this was largely a function of having more memory available. Budge added it was also a function of people getting better at programming. Five years ago, he said there weren't that many people writing computer games. Now there were thousands of people capable of writing a good video game. And a few designers had "upped the ante" to the point where it was not enough just to write a good game–you also had to include extra features like a game editor.

Hawkins then demonstrated the final program for the episode, *One-on-One*, a basketball game featuring the likenesses–well, I use that term loosely–of NBA players Larry Bird and Julius "Dr. J" Erving. Bird and Erving actively participated in the design process, according to Hawkins. Hawkins said there were photo sessions with both men to help "capture them in action." EA designers talked to both players about "their moves," and combined that with statistics–notably, the players' shooting percentages from different parts of the court–to create the player models for the game.

Cheifet asked Hawkins about the "next generation" of computer games. Hawkins said at a simple level, the goal was to extend the realism and graphics seen in EA's current products. He also pointed to "more creative possibilities for the player," by extending concepts like PCS. In fact, EA had already launched *Music Construction Set*, which used a lot of Budge's ideas from PCS. The greatest challenge moving forward was looking for original ideas that "broke the mold."

Kildall ended the program by asking where those new ides came from. Did people just come to EA with them? Hawkins said it happened in many different ways. For instance, EA's recent platformer *Hard Hat Mack* came from two high school students (Michael Abbott and Matthew Alexander) who drew up the idea on paper during their physics class. Hawkins said he signed a contract for the game based on that alone.

## Tax Software Is Hot This Month!

Stewart Cheifet presented this episode's "Random Access" segment, which is dated around the end of February 1984.

- Hewlett-Packard announced its new $500 portable printer, a non-impact inkjet model with low noise, battery paper, and the ability to print 150 characters per second.
- IBM announced a licensing deal with Intel to allow the PC manufacturer to make its own 8088 microprocessors. Rising demand on Intel had prevented IBM from producing enough PCs to meet market demand. IBM also obtained the rights to make its own 80286 processors for its planned next-generation model of the Personal Computer, currently nicknamed "Popcorn."
- There were reports that United Kingdom-based Applied Computer Technologies would buy Victor Technologies and move the latter's manufacturing operations from Silicon Valley to Scotland.
- Tandon Corporation, the largest manufacturer of disk drives, said it would lay off 1,000 employees and transfer its manufacturing operations overseas to cut costs. Cheifet noted this was part of a trend–in the past year, at least 20 countries had sent delegations to Silicon Valley to lure business abroad. Of those nations, Ireland had proven the most successful.
- The Japanese Diet was debating a bill that would limit copyright protection for U.S. software to 15 years.
- Cheifet said the "hot software" this month was tax preparation software, with 2 of the the more than 20 programs on the market–Continental's *Tax Advantage* and Howardsoft's *Tax Preparer*–breaking into the top-10 for the week.

- Computer magazines continued to boom, with Cheifet noting there were more than 300 now on the market. The fastest growing publications were *Compute!*, *Personal Computing*, *Popular Computing* (the sponsor of "Random Access"), and *InfoWorld*.
- Finally, Cheifet joked the last place you'd want to see a robot was a bar. But this past week in San Francisco, the world's first robot bartender made its debut. The robot could talk, take orders, and mix 200 different drinks. Unfortunately, when a waitress asked for a Bloody Mary and a beer, the robot knocked a glass onto the floor and started pouring beer all over the counter. The robot's designer acknowledged there were still some bugs to work out.

### Paul Schindler on Communications Software

This "Random Access" segment also featured the first of many quick software reviews by Paul Schindler, a technology journalist then affiliated with the *Whole Earth Software Review*. In this vignette, Schindler actually provided an overview of communications packages. He noted that fewer than one-fourth of all personal computers could actually communicate with other machines. So he recommended the following software packages for those "looking to take the plunge":

- for users of CP/M-based machines, MITE and MODEM7;
- for IBM Personal Computers, SMARTCOM 2, CROSSTALK 16, or PC TALK 3 (the latter being the cheapest at just $35);
- for Apple users, TSC Terminal (for 40-column screens) or Data Capture 2 (for 80-column screens); and
- for Commodore & Radio Shack computers, VIDTEX.

## Stevens Caught in Apple Crossfire

The three "superstar" game designers featured in this episode all had interesting post-*Chronicles* careers. Jessica Stevens, however, made her mark largely outside of gaming. At the time she designed *Space Shuttle*, she ran her own company, Woodside Design Associates, Inc., which described itself as a "high technology think tank." Stevens basically developed her game as an outside consultant for Activision. She also ported the game *Carnival* to the Atari 2600 for Activision.

Stevens and Woodside Design's real focus was developing flat-panel display technology. This led to the demise of Woodside in the late 1980s. Stevens had struck a verbal agreement in February 1985 with Steve Jobs to have Apple buy Woodside for $5 million and then spend another $20 million to develop flat-panel displays for a "book-size Macintosh" project that Jobs was pushing. Unfortunately, the day the deal was to be signed, Jobs and Apple President John Sculley's infamous feud came to a head, which ended with the company's board firing Jobs. Sculley then abandoned Jobs' plan to develop a Mac with a flat-screen and scuttled the deal with Stevens. Woodside Design then filed a $1 billion breach of contract lawsuit against Apple. I couldn't find any information on the outcome of that suit, but I'm guessing it never went anywhere.

Stevens founded a new company, Telegen Corporation, in 1990, which also focused on flat-panel display technology. The company went public with Stevens serving in various executive roles up until the company's bankruptcy in the early 2000s. The last corporate filing from Telegen, with Stevens serving as CEO, was filed with the SEC in January 2003.

## Crawford Rejected Computer Games for Lack of Artistic Merit

Chris Crawford developed a number of games while working at Atari from 1979 to 1984. His most notable project was *Eastern Front (1941)*, a turn-based wargame for the Atari 8-bit computer line. After that game became a best-seller, Atari promoted Crawford to manage its games research group, during which time he developed *Excalibur*.

The North American video game crash of 1983 led Atari to lay off a good chunk of its workforce, including Crawford, who went out on his own as an independent developer. He created *Balance of Power*, a Cold War political simulation, for the Macintosh in 1985. Crawford said this was his most commercially successful game, selling 250,000 copies.

But commercial success was not Crawford's main objective. He cultivated a reputation in the late 1980s and early 1990s as a fierce critic of the games industry, arguing that designers should promote gaming as a medium of artistic expression rather than entertainment. Crawford founded and published *The Journal of Computer Game Design* from 1987 to 1996, which featured a number of editorials in which he spelled out his problems with the industry. For instance, in the October 1991 issue, he argued:

> Art is an expression by the artist; entertainment is an expression for the audience. The artist is primarily concerned with himself; the entertainer is primarily concerned with his audience. The artist seeks to express the truth within himself; the entertainer seeks to give the audience what it wants.
>
> Entertainment is therefore intrinsically commercial and art is just as intrinsically noncommercial. You can sell entertainment and you can't sell art. (It's true that snobs will pay a lot of money for your art, but only after you're dead.)
>
> Having said this, I can now state that almost nothing in the computer game industry comprises art. We are an entertainment business.

Famously, Crawford delivered what became known as the "Dragon's Speech" at the 1992 Computer Game Developers Conference, an event that he had created in 1988. This speech marked Crawford's exit from game development in favor of what he dubbed the next generation of "interactive storytelling." To that end, Crawford changed the name of his publication to *Interactive Entertainment Design* in 1993 and began work on a new game engine, Storytron, and an associated game, Siboot. But after more than two decades, he failed to make much progress with either. Crawford later explained why he was giving up on interactive storytelling:

> My goal all along has not been to create a single great piece of interactive storytelling; it has been to ignite a new industry, a community devoted to the furtherance of interactive storytelling as an art form. Suppose that I could indeed finish Siboot to my own satisfaction. What would happen?

> First, few people would appreciate it. They would judge it by the only standards with which they are familiar: the standards of current game design. They would conclude that Siboot is a lousy game — and they would be right. Siboot was never intended to be a game; it's interactive storytelling. The emphasis is on character interaction, and it already offers interesting dramatic character interaction. But people aren't looking for interesting dramatic character interaction; they're looking for the things that make great games: challenge, a smooth learning curve, impressive graphics, catchy little tunes to accompany their play. Above all, a game must be winnable. Yet stories aren't necessarily about winning and losing; they're about drama.

## Budge Went from Construction Sets to Building Web Browsers

Our final designer, Bill Budge, seems to have had a much less dramatic career than Crawford. Budge never did make the more abstract construction set he mentioned to Setwart Cheifet. But he did go on to write *MousePaint*, a popular drawing program for the Apple II.

Budge later joined Trip Hawkins' other venture, The 3DO Company, where he served on the engineering team for nine years. Budge then returned for a brief stint at EA in 2003, working on the company's game based on *The Godfather*. After that, Budge worked at Sony. He joined Google in 2010 as part of the Chrome web browser team. In January 2022, Budge announced his retirement from Google.

Like Budge, Trip Hawkins worked at Apple prior to the latter founding Electronic Arts in 1982. Hawkins remained at the helm of EA until 1991, when he started 3DO, which marketed its own video console under that same name from 1993 to 1996. When the 3DO console failed to take hold, the company transitioned to simply developing games. Hawkins remained CEO until the company's bankruptcy in 2003. That same year, Hawkins founded Digital Chocolate, a mobile games development company, which lasted until 2014.

## Pushing the Limits of 1984 Gaming Capabilities

*Space Shuttle* and *Excalibur* were both fairly sophisticated games for 1984. Dan Persons' review of *Space Shuttle* in the April 1984 issue of *Video Games* went so far as to say this was "not a game," but a "full-fledged, strikingly accurate simulation of a mission into space." Persons noted *Space Shuttle* was "mind-bogglingly hard," but that was probably because "space travel is hard." It was also not a quick game, as Persons observed it took him a "full half hour" to dock the space shuttle with the orbiting satellite, and "thirty minutes to dock with two more."

*Excalibur* reviewers were also quick to note that game's difficult learning curve. According to David Duberman's review in the February 1984 issue of *Antic*, *Excalibur* was "perhaps the most complex [game] yet written for an eight-bit computer!" It was "impossible to play *Excalibur* with the minset you might use for a game of Pac-Man, for example." Duberman noted the game came with a 100-page manual, which included a 69-page novel that the player would probably need to read multiple times to understand and complete the game's objectives.

As Crawford mentioned during his *Chronicles* segment, the large size of *Excalibur* required switching between multiple disks. David P. Townsend explained in his review for *Computer Gaming World*:

> The game itself is divided into three modules, *Camelot*, *Britain*, and *Battle*. The importance is that only one can be in memory at a time, so when you move from one module to another you must wait for the computer to swap the current module out to disk and bring the new module into memory.

Crawford himself included a "designer's note" to accompany Townsend's review. Perhaps not surprisingly, he is equally complementary and critical of his own creation. While saying he felt "deep pride in this game," he nevertheless found it was "still too 'wargamy' for my taste." He also lamented the lack of commercial success, complaining that his "magnum opus" had "not won the attention I think it deserves."

## Larry and Dr. J Make EA a Sports Powerhouse

In contrast to the more esoteric offerings from Stevens and Crawford, the two Electronic Arts games proved to be much more accessible commercial hits. Elaine Holden praised *Pinball Construction Set* in the January 1984 issue of *Byte* magazine, noting, “It’s really hard to find anything wrong with this game.” She cited Budge’s “marvelous sense of programming” and credited him with creating a “classy game” that was simple to grasp yet still fun “for even the most advanced player because of the challenge and the endless variety of modifications, refinements, and degrees of difficulty available.” And unlike *Excalibur* and its novel-in-a-manual, PCS only came with a 13-page users guide.

*One-on-One* enjoyed similar praise. A review in the March-April 1984 issue of *St. Game* declared it “*the* sports game of 1984.” The reviewer said that “[e]ven nonathletes have a fair chance at beating the computer or another player, but knowing a little about real-life basketball doesn’t hurt,” given the game’s accounting for the real Bird and Erving’s “respective strengths and weaknesses.” The game’s designer, Eric Hammond, was also described as a “formidable basketball player himself.”

The game itself was a landmark in that it was the first to feature real-life professional athletes. Patrick Sauer, in a 2017 retrospective for *Vice*, noted “the game was a huge financial boon for EA Sports”–selling over a million units according to Trip Hawkins–“but the biggest payoff was in how it shaped the company.” Indeed, Bird and Dr. J helped pave the way for another, more lucrative EA sports franchise–the *John Madden* football games.

## Notes from the Random Access File

- This episode was recorded on February 28, 1984, and is available at the Internet Archive.
- *Dragon’s Lair* was directed by famed animator Don Bluth, perhaps best known for his 1982 film *The Secret of NIMH* . If you’d like to learn more about Bluth and the *Dragon’s Lair* game franchise, I recommend this episode of the Don Bluth-focused podcast *The Bluth, the Whole Bluth, and Nothing But the Bluth*, hosted by Dax Schaffer and Sara Iyer.

- Jessica Stevens' brothers, Garry Kitchen and Dan Kitchen, are actually better known in the world of game design. The Kitchen brothers worked for Activision in the early 1980s, where they created and ported a number of games for the Atari 2600. They both continue to work in the field today.
- EA apparently only released one other game around this time with "Construction Set" in the name: *Stuart Smith's Adventure Construction Set*.
- According to the *Vice* article, Julius Erving and Larry Bird each received $25,000 for participating in the design of *One-on-One* as well as 2.5 percent royalties on sales; Erving also received some EA stock.
- The IBM "Popcorn" would be released in August 1984 as the IBM PC AT.
- Applied Technology did not end up buying Victor Technologies. Instead, Sweden-based Datatronic AB purchased 90 percent of Victor, which was in Chapter 11 bankruptcy, in August 1985.
- The robot bartender mentioned by Cheifet actually ran off of an Apple II computer.
- Although Cheifet noted there was an ongoing boom in computer magazines, the *Whole Earth Software Review* that employed Paul Schindler was not among the more successful ones. It only produced three issues in 1984 before merging with another publication under the name the *Whole Earth Review*, which published quarterly until 2003.

# Chapter 16: The Apple Logo Programming Language

Today, Python is probably the most popular computer programming language taught in elementary and secondary schools. (There's even a terrific podcast, *Teaching Python*, on this subject.) But back in the 1980s, BASIC was the language of choice for many introductory computer classrooms. Specifically, versions of Microsoft BASIC came with many popular 8-bit microcomputers, including the Apple II and Commodore 64, which were also commonly used in schools at the time.

However, BASIC was not the only educational programming language of the 1980s. There was also Logo, a language first developed in the late 1960s and modeled on an even older programming language, Lisp. Logo was especially popular in schools because of its use of graphics. This next episode of *The Computer Chronicles* provides an overview of Logo as part of a broader discussion into the use of computers in education.

## The Computer Terminal as "Private Tutor"

Stewart Cheifet and our old friend Herb Lechner opened this episode by demonstrating an educational game, *Algebra Arcade* by Wadsworth Electronic Publishing Company, on an IBM Personal Computer. Cheifet explained the game was designed to teach students algebraic expressions and their graphical representations. The game presented the player with a pair of axes and a number of "algebroids" positioned throughout the graph. The test for the player was to write an equation using algebraic functions to try and plot a graph that would intersect as many of these algebroids as possible. There was also a "graph gobbler" the player needed to avoid. If the final equation hit the gobbler, the player received a penalty.

With respect to the broader subject of computers in education, Cheifet pointed out that some people were afraid of technology and feared that computers would replace teachers and dehumanize education. Did Lechner see that in the future? Lechner noted that the expectations for the use of computers in schools was very high. But in practice, it had not been nearly as extensive as expected, and he did not think the positive or negative impacts so far had been terribly significant.

Cheifet then narrated our usual B-roll, this time taken at the Lucille M. Nixon Elementary School in Palo Alto, California. Cheifet said that with over one-half of U.S. schools now equipped with computers, students from elementary school through college were likely to have some computer instruction at every level of their education. But there were still doubts over the best way to integrate computers into a traditional curriculum.

Replacing the teacher with a computer terminal changed the focus of learning away from the instructor and to the computer. While this diminished traditional contact between the student and teacher, the terminal became a kind of "private tutor," adjusting to the different learning rates and abilities of each student. The computer was also interactive, demanding answers within a specific length of time, making corrections, and helping the student to understand the reasons behind the mistake.

Cheifet explained that programming languages for children relied on colorful graphics and game-like approaches to teach children analytical skills and problem-solving logic. This meant that for younger children, time spent at the terminal could seem more like a game than a lesson.

The primary use of computer instruction was for rote learning and practice drills, Cheifet said. But there was another level of applications that demanded much more creative interactions. Programming languages like Logo were designed to teach children to make up their own programs, create their own commands, and in general put students in the active role of controlling the machine instead of just responding to an electronic inquisitor. Cheifet said proponents of this approach claimed that students trained to think logically would become better at problem solving in general. But at least one study questioned whether there was any difference between a Logo-trained and a traditionally trained child.

# A 1.5-inch Rolls Royce That Only Costs $2.50!

For the first guest segment, Cheifet and Lechner welcomed Professor Patrick Suppes of Stanford University and Nancy Palmer, the computer education coordinator for the Palo Alto School District. Lechner repeated his earlier observation that the application of computers in education was not moving along as rapidly as everyone had hoped. How did his guests feel about that assertion? Suppes said that for most of the past 20 years that he'd been involved with computers and education, he agreed that was true. But in the last 3 or 4 years, he said the pace had really changed. There was now a lot of hardware and an increasing amount of software in the schools.

Palmer concurred, at least with respect to her school district. She said there was now a lot more interest from the teachers, and they were starting to use computers within their classroom in a lot of different ways. Students were using word processing and some other types of programs like data management, making the computer more of a tool. Of course, students were also learning programming languages like BASIC, Logo, and Pascal. But there was now a wider application of computers. Suppes emphasized that we had only really begun to integrate computers into education.

Lechner asked what triggered this movement of the past few years. What was different now versus 10 years ago? Suppes said it was primarily the enormous reduction in the cost of hardware. Palmer added it was also changes in the technology itself, notably the creation of the microcomputer market over the past eight years. That made computers a lot more reasonable for schools to buy and use. And as the technology improved, so did its ease of use. Suppes repeated a joke he'd heard that if cars followed the same trail as computers, a Rolls Royce would cost $2.50 and be just 1.5 inches long.

Cheifet then asked Suppes about his own work using computers to teach his introductory logic course at Stanford. How did that work? Suppes explained that he did not give any lectures for the course. Nor were there any assigned textbooks. Instead, students used computer terminals to review course materials and complete exercises. The students were free to come and go as they wished 24/7, although a teaching assistant was available every

Saturday night from 8 to 11 p.m. Suppes said this provided many advantages from the student's standpoint.

Cheifet asked about the instructor's standpoint. Was this actually providing a better education? And did you pay a price for this type of convenience? Suppes said he didn't think anyone was paying a price, but of course there was always some give-and-take with any approach to instruction. In this case, it was not just about the freedom the students had to come and go, but also the individualization. Some students just learned the topic much more easily and quickly than other students. With computer-based instruction, this individuality was taken into account, i.e., the pace of one student did not affect the pace of another student, which was not the case in a traditional classroom setting.

Suppes also emphasized his approach afforded students excellent opportunities to practice–and be corrected–in performing logic exercises and giving proofs and counter-examples. Suppes then provided a demonstration of his actual course using a terminal tied into the Stanford mainframe. Suppes noted it took a student about 75 hours to complete the entire course.

After the demonstration, Cheifet turned to Palmer and asked her if high schools were using computers to teach actual classes like this, or if they were just used to teach programming. Palmer said they were used in both ways in Palo Alto. For example, business classes used computers to teach accounting. Lechner asked if the computers made any difference. Were the achievement scores going up for the classes using computers? Palmer said that was difficult to answer in terms of real data. She felt there had been a difference with the students, but they were not at the stage where hard data has been collected.

## "It Seems a Bit Whimsical. What's the Big Deal?"

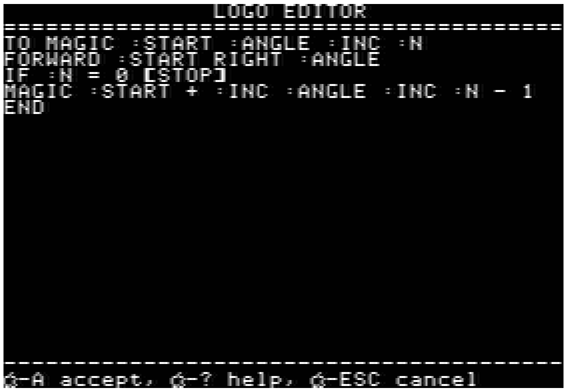
For the final segment, Glenn Kleiman joined Cheifet, Lechner, and Suppes. Kleiman owned Teaching Tools, Inc., and had recently authored a book on computers in education, *Brave New Schools*. Cheifet asked Kleiman to talk about the significance of the Logo programming language. Kleiman said

Logo had been widely touted as the language to introduce kids to computers, even over BASIC. He added there had been a lot of other claims made about Logo, such as that it was an ideal way to teach problem-solving skills, although he did not necessarily buy into that hype. Nevertheless, he felt Logo had a number of advantages for introducing kids to programming, and it did have some implications for more general problem-solving skills.

Kleiman was there specifically to demonstrate Apple Logo on an Apple II. (As with BASIC, there were a number of different implementations of Logo targeting specific machines.) Kleiman explained there were actually two main components of Logo. The first, which was very well known, was *turtle graphics*, which was widely used with children. The other component involved list processing, which enabled Logo to work with words, sentences, and other types of things. But it was really turtle graphics that most students associated with Logo.

Kleiman explained that with turtle graphics, you could instruct an on-screen pen–the “turtle”–to move forward, turn right or left by a number of degrees, and mark its trail to create graphics. Turtle graphics allowed students to take these very simple commands and combine them into more complex programs called *procedures*. Kleiman demonstrated a sample procedure called “MAGIC” that he had created.

Rather than just try and describe all this–or display blurry screenshots from the episode–I went ahead and installed an original version of Apple Logo on an Apple IIe emulator and recreated Kleiman’s MAGIC demo. First, I typed the MAGIC procedure into the built-in Logo Editor:



```
               LOGO EDITOR
========================================
TO MAGIC :START :ANGLE :INC :N
FORWARD :START RIGHT :ANGLE
IF :N = 0 [STOP]
MAGIC :START + :INC :ANGLE :INC :N - 1
END
```

A screenshot of the Apple LOGO Editor showing the sample code listed below.

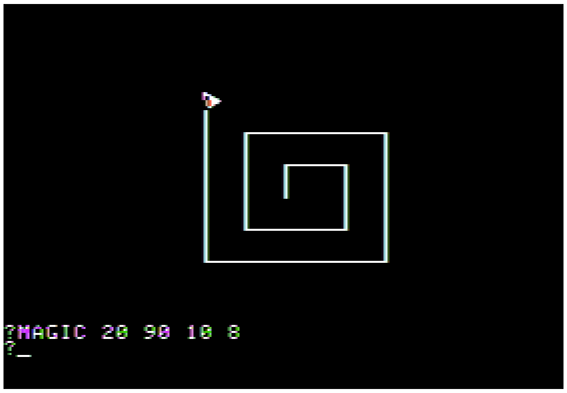
Here’s a printout of the source code:

```
TO MAGIC :START :ANGLE :INC :N
FORWARD :START RIGHT :ANGLE
IF :N = 0 [STOP]
MAGIC :START + :INC :ANGLE :INC :N - 1
END
```

Basically, this procedure creates a new word–MAGIC–that Logo can recognize. The items preceded by colons are variables. The user invokes MAGIC by entering four variables–start, angle, increment, and *N*. The procedure then moves the turtle forward by *start* number of steps, turns the turtle to the right by *angle* degrees, and then moves the turtle forward again, this time by adding the *increment* to the previous *start*. This sequence is then repeated for *N* number of turns.

These are the examples presented by Kleiman that I recreated:

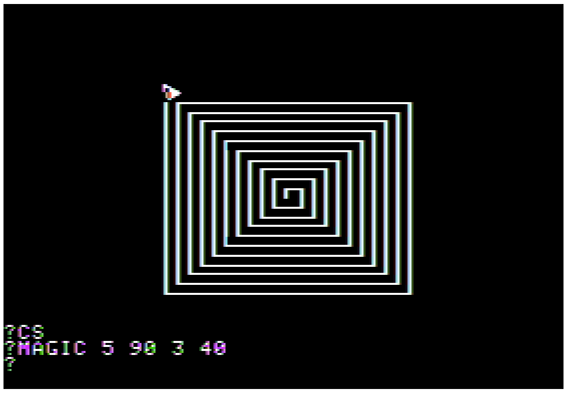
```
? MAGIC 20 90 10 8
```



```
?MAGIC 20 90 10 8
?_
```

A spiral created using the Apple LOGO command “MAGIC 20 90 10 8.”

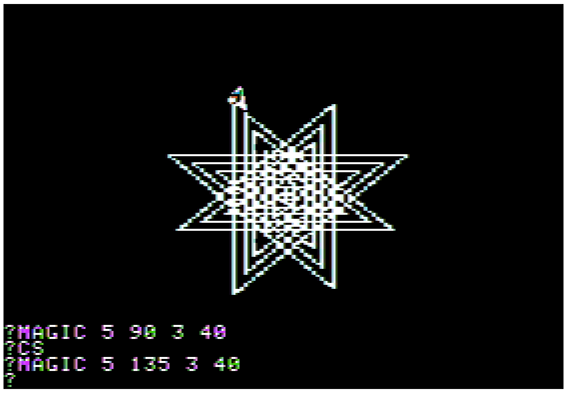
```
? MAGIC 5 90 3 40
```



```
?CS
?MAGIC 5 90 3 40
?
```

A spiral created using the Apple LOGO command “MAGIC 5 90 3 40.”

```
? MAGIC 5 135 3 40
```



```
?MAGIC 5 90 3 40
?CS
?MAGIC 5 135 3 40
?
```

A star created using the Apple LOGO command “Magic 5 135 3 40.”

Cheifet snarked that while this was kind of interesting, and it was fun to see these pictures, “it seems a bit whimsical. What’s the big deal?” Kleiman conceded it was perhaps whimsical in a way. But kids enjoyed whimsy. He thought the “big deal” was that it was a very different approach that what Professor Suppes had demonstrated earlier. Suppes used computers to do things that had always been done in education–tutorials and drills. And that was certainly a valid use of the technology, Kleiman said. What Logo provided, however, was an example of introducing a tool that kids could explore, where the child had more control than the computer. This was not about problems and questions coming out of the computer–the computer was waiting for the child to act.

Cheifet asked Suppes for his view of this use of computers versus his work. Suppes said he agreed with Kleiman that the use of Logo could be valuable and fascinating for young students. The nice thing about Logo was that it gave students a quick and easy sense of how a computer language worked.

Cheifet wanted to know if there were social–or perhaps anti-social–consequences to coupling kids with machines rather than traditional “flesh and blood” teachers. Suppes said it was important to consider different use cases. In the elementary school it was very important for the student to have a personal relationship with the teacher. Indeed, one of the paradoxes of American education was that while the data overwhelmingly showed this relationship was critical, class sizes on average were larger in elementary schools than in secondary schools. But as students became adults, Suppes said they were in a different kind of world. He said many adults in the future might want to get a good deal of their education at home over a terminal, because it was inconvenient to go to a campus site.

Lechner followed up, asking if in the future he thought it was possible an entire curriculum could be taken by a mature student over a computer. Suppes said sure, and his own courses at Stanford were meant to serve as an example. The socially important thing was teaching courses this way allowed for decentralization. It was really a social, administrative, and psychological decision rather than a technical decision.

Kleiman said he took a somewhat different view. He saw computers mainly as a tool for the teacher to use. He did not think computers were capable of replacing teachers. Kleiman noted he actually took Professor Suppes’ logic class the first year it was offered by computer. Kleiman said he thought the computer provided a good means for practicing logic, but there was an intelligence and intuition to the subject matter that one could get from a professor of Suppes’ caliber better than any computer program.

# Patirick C. Suppes (1922 - 2014)

Patrick Suppes passed away in November 2014 at the age of 92. According to his *New York Times* obituary, Suppes joined Stanford University’s philosophy department in 1950 and remained with the school for the next

64 years until his death. During his extraordinary tenure, the *Times* noted, Suppes "influenced a wide range of academic fields, among them logic, physics and psychology."

Suppes was one of the earliest promoters of using computers in education. In a 1966 paper, the professor discussed his earliest experiments into computer-assisted instruction. One such experiment involved "tutorial sessions" for young children, who used a light pen with a CRT monitor to learn basic vocabulary:

> With such a tutorial system we can individualize instruction for a child entering the first grade. The bright child of middle-class background who has gone to kindergarten and nursery school for three years before entering the first grade and has a large speaking vocabulary could easily finish work on the concepts I have listed in a single 30-minute session. A culturally deprives child who has not attended kindergarten may need as many as four or five sessions to acquire these concepts. It is important to keep the deprived child from developing a sense of failure or defeat at the start of his schooling. … It is equally important that a tutorial program have enough flexibility to avoid boring a child with repetitive exercises he already understands.

Closer to the time of his *Chronicles* appearance, Suppes presented another paper, "Computers and Education in the 21st Century," which spelled out his long-term vision for what the world would look like in 2084. He continued to iterate on the importance of individualization and predicted that computers would eventually make traditional secondary schooling obsolete:

> Schools will be close to where adolescents are living. Teachers will be like tutors, but not like the tutors in ancient Athens in the sense that that they will not be carrying the full load of the instruction. Teachers in secondary schools will be counselors and trouble shooters, not just counselors about future careers or emotional problems, but cognitive counselors, able to help when a student has not found the right mix or the right approach to instruction in the wealth of technological offerings. The demands on the teachers will be greater than they are now. They will need to be more professional and more deeply trained,

> and they should be paid much better than they are now. The expert teachers I see as needed in the technological setting of the secondary school in 2084 must be technically expert and intellectually sophisticated. They should be compensated accordingly.

## Logo or BASIC?

Glenn M. Kleiman also found a home in academia. As mentioned above, he attended Stanford and took Professor Suppes' logic class in the 1970s. After receiving his doctorate in cognitive psychology from Stanford in 1977, he founded the aforementioned Teaching Tools company in 1981. After four years of independent consulting and software development, Kleiman joined the non-profit Education Development Center, where he served in various roles over the next 22 years. During this time, he also lectured at Harvard University's Graduate School of Education.

In 2007, Kleiman joined North Carolina State University as the executive director of its Friday Institute for Educational Innovation. He stepped down as executive director in 2018 but remains on faculty as a professor.

As his *Chronicles* appearance illustrated, Kleiman was a cautiously optimistic supporter of Logo. In an article from the July 1982 issue of *Compute!*, he compared the relative strengths and weaknesses of the newer Logo versus the older BASIC. Kleiman was measured in his praise for the latter, however, noting that many Logo advocates insisted it was "the only proper approach, and that teaching BASIC is almost evil." Kleiman argued it was not that cut-and-dry:

> While the turtle graphics component of Logo suits children very well, I am less certain about the rest of Logo. The commands for working with words are powerful but may not be simple for children to use. Most teachers using Logo have so far focused on using turtle graphics. I am waiting for more information from teachers and children who are exploring the non-graphics aspects of Logo.
>
> In BASIC, children generally find it easy to understand the individual string processing commands and to write small programs with them.

> However, the difficulty of BASIC programming increases rapidly as the size of the program increases.

Kleiman’s conclusion at the time was that it did not matter *which* language children learned. The more important question was “whether the teaching encourages careful task analysis, problem solving, and testing.” In that sense, he thought Logo, which encouraged “modular programming,” had the advantage over BASIC. He added that turtle graphics also had the side benefit of teaching children about geometry and symmetry.

## Notes from the Random Access File

- This episode is available at the Internet Archive and was recorded in late 1983 or early 1984. I could not nail down the date with any greater accuracy.
- Much like BASIC, there were several different implementations of Logo, even within the same platform. For instance, Kleiman’s *Compute!* article noted there were at least two distinct versions available for the Apple II: one by Logo Computer Systems, Inc. (LCSI), and the other by MIT. Apple marketed LCSI Logo, while two outside companies sold the MIT version. (In my emulation re-creation, I used Apple Logo II, an LCSI release from 1984.)
- According to Kleiman, the various Logo packages sold for between $169 and $179 as of late 1982. You could also buy a program called *Nibble*, which provided just a simplified version of the turtle graphics package, for around $30.
- The only commercial software ever produced by Teaching Tools, at least according to Moby Games, was Word-a-Mation, an educational game written by Kleiman for the Apple II.
- Unfortunately, I was unable to track down any verifiable information regarding Nancy Palmer or her current whereabouts.
- The American Philosophical Society named an award for Professor Suppes. The Patrick Suppes Prize is actually a multi-disciplinary award that rotates its recipients among the fields of the philosophy of science, the history of science, and psychology. This year’s recipient (for the history of science) is Professor Jessica Riskin of Stanford University.

- I mentioned Python as the successor to Logo when it comes to teaching today’s students about computer programming. There is actually a turtle graphics package available as part of the Python Standard Library. Episode 10 of *Teaching Python* podcast, presented by Sean Tibor and Kelly Paredes, does a deep dive into this modern-day reimplementation of Logo.
- To be honest, I never knew about the non-turtle graphics part of Logo until I watched this episode. Like many elementary school computer students in the late 1980s, we were mostly taught BASIC with the occasional detour into Logo’s turtle graphics.

# Chapter 17: Alan Shugart and the Memorex 3680

This next episode of *The Computer Chronicles* from 1984 is about storage devices, specifically disk drives. At this point in the microcomputer revolution, the 5.25-inch floppy disk is the accepted standard. But a number of new technologies are vying to supplant it. And while Gary Kildall was bullish on at least some of these new technologies eventually gaining mainstream acceptance, one of the guests aggressively pushed back on the idea.

## A Brief History of Computer Storage Devices

Stewart Cheifet opened the episode by noting that while disk drives were still considered a “peripheral” device, it was really a key part of any modern computer system, since you could not run many serious applications without a disk drive. Kildall agreed, adding that the traditional model of a computer included a backup storage drive, a CPU, and main memory. The scaling down of all three of these components is what led to the microcomputer revolution in the first place. For example, semiconductor-based memory chips started to replace core memory in the early 1970s. But it wasn’t until floppy disks that we had a complete computer system that was useful to ordinary people.

Kildall said it was also interesting that the disk drive remained the only key component of the computer that still relied on mechanical parts. He was interested to see if that would continue or if disk drives would start to become more semiconductor-based.

Cheifet then narrated our B-roll for this episode, providing a brief overview of mass storage technology. He said progress in the field of mass storage was directed by three needs: volume and density of information, and the speed of accessing that information. In the 1950s, computer storage borrowed the recording technology used by sound engineers, i.e., magnetic tape. Like its earliest predcessor, the wire recorder, modern oxide tapes stored data sequentially. So to reach information stored near the end of a reel, the user had to run through everything on the tape up to that point. Short of developing faster tape drives, the only way to make the data more accessible was to change the format of the recording medium itself.

Disk storage, which allowed for *random access* to information, was already in development in the mid-1950s, Cheifet said, but the popularity and potential of this technology had only become clear in the past decade. The original 24-inch disk platters had now shrunk to just 5.25 inches, while the accompanying bit-density had increased 3,900 times, from a few thousand to nearly 8 million bits of data per square inch of disk. Cheifet said the improvements that led to these advances had now given way to a new approach promising yet another dramatic increase in density. Thinner film heads and tinier components could read bits recorded much closer together on narrower tracks. Another promising development was "perpendicular recording," which arranged tracks on a magnetic disk vertically rather than horizontally. Cheifet said this small change alone could double or triple a disk's storage capacity.

## 5.25-Inch Floppy Disk Pioneer Sees No Threat to Continued 5.25-Inch Floppy Disk Dominance

Alan Shugart, the founder and CEO of Seagate Technology, joined Cheifet and Kildall in the studio. Seagate was the leading manufacturer of 5.25-inch "mini-floppy" disk drives. Kildall noted he got his first drive from Seagate for free some years earlier. Shugart quipped they didn't give anything away anymore. Kildall then lobbed a softball question, asking Shugart if he believed the floppy disk drive was instrumental in setting off the microcomputer revolution. Shugart said there was no doubt in his mind that was the case. The original 8-inch floppy drives had proven the technology could work, while the 5.25-inch mini-floppy drive expanded the market and provided for the "big explosion" in personal computers.

Kildall noted that Seagate was now producing "Winchester" drives, i.e., hard disk drives. How would that affect personal computers? Shugart replied that the hard drive was complementary to the floppy disk drive and helped to expand the capacity of a small desktop PC to handle jobs that required a lot of storage. After all, you didn't see any smaller computers with hard drives that did not also include at least one floppy drive. Kildall pointed out the floppy drive provided a means of transferring data from one computer to another. Shugart added that multiple drives also provided backup.

Kildall said that one criticism of using hard disks with small computers was the concern about breaking the mechanical parts when moving the drives around. Was Shugart concerned about the reliability of hard disk drives? Shugart agreed the hard drives were more susceptible to damage from shock or vibration versus a floppy drive. The hard drive's magnetic heads had to stay in contract with the media at all times, unlike a floppy drive where you could remove the diskettes to protect them from damage. So in that sense, the Winchester drives were more delicate. Kildall asked if this problem would be addressed in the future. Shugart said newer versions of the drivers would be more shock resistant.

Kildall then showed an example of a 5.25-inch floppy drive. He noted it still had a lot of mechanical components inside, making it akin to the "Model T" of computing. Then Kildall produced an example of non-mechanical storage technology, Intel's bubble memory module, which could hold 1 MB on a single chip.

Kildall said some people–presumably including himself–believed that some form of semiconductor-based storage like bubble memory would eventually replace the floppy disk drive. Did Shugart agree? He emphatically did not. Shugart noted the floppy disk drive had a theoretically unlimited storage capacity, because the user could keep removing the disk and putting in a new one. Bubble memory could not offer this. And on a cost-per-bit basis, Shugart believed you would never be able to beat magnetic disk storage with a semiconductor device.

Cheifet then pointed to the fact many personal computers now came with a 3.5-inch (or "micro-floppy") floppy disk drive. Could Shugart see this smaller size replacing the traditional 5.25-inch disks? Again, Shugart said no. While he conceded there was a market for the smaller size–and, he noted, there were actually a number of competing micro-floppy standards ranging between 3 and 3.75 inches–none would ever replace the 5.25-inch mini-floppy. The main reason, he said, was that most of the world's software was written on 5.25-inch disks and you would never "in your wildest imagination" transcribe all of that software onto smaller disks. He said it just would not be done.

Kildall then produced yet another example (see image below) of a possible competitor to the floppy disk, the video disc. Specifically, Kildall displayed a Capacitance Electronic Disc (CED), which was a non-optical video disc format produced by RCA. Kildall said one CED could hold about 1 GB of data, enough space to contain the entire *Encyclopedia Britannica*. Did Shugart see any potential for *this* type of format in PC storage? Not surprisingly, Shugart said no, except for possibly as archival storage. He said the main problem with video disc technology was that there was a substantial error rate. With an audio or visual recording–which is what CED's were used for–that wasn't a big deal since a human could not perceive minor imperfections. But this was not good enough for business computing, which required a much more reliable storage medium like floppy disks.

Cheifet followed up, asking about the possibility of using optical laser-based–discs for storage. Would that be more reliable? Shugart reiterated his view that such discs would not be useful for anything beyond archival storage. And as far as optical storage that had the capability of updating in the same manner as magnetic storage–Shugart said that was still in the "laboratory curiousity stage."

Cheifet switched subjects, asking about the perpendicular recording technology he mentioned in the B-roll. Shugart briefly explained how the idea worked. Basically, when you stored data on the surface of a disk, you "flipped" little magnets that were lying on their sides. And the drive then later detected whether or not the magnets were flipped. There were only so many magnets you could get on the surface–i.e., the recording density. But if you could somehow get those magnets standing "on end," like a cord of wood, then you could get more magnets onto each surface. Shugart said this type of vertical or perpendicular recording was definitely in the cards as a future technology.

Finally, Cheifet asked about double-sided or “double density” 5.25-inch floppy disks. He noted there was some controversy as to whether double-sided disks were as reliable as traditional, single-sided disks. Shugart said there was no “fundamental reason” that double-sided disks should not be as reliable. Perhaps some floppy disk vendors had “stretched the technology” further then they should, and that had unfairly given double-sided disks a bad name.

## Memorex Exec Touts “Thin” Technology

For the final segment, Frank Sordello of Memorex joined Cheifet, Kildall, and Shugart to talk about his company’s newest hard disk drive, the Memorex 3680. Sordello did not bring the actual drive–which came in a large cabinet, as was the style at the time. Instead, he showed one of the nine 14-inch disk platters that came with the drive. Sordello said that when all nine platters were assembled, the drive could hold up to 1.2 GB of data. He also showed some of Memorex’s smaller 8-inch and 5.25-inch hard drive platters.

Sordello explained a key feature of the 3860, its thin-head film technology. Traditionally, Sordello said, hard drive platters used heads that were like a “bent nail” wrapped in wires. In order to lay down 150,000 bits-per-inch of data along the circumference of a hard drive platter, you needed a very small nail. Over the past eight years, those heads were made using a mechanical grinding process. But thanks to advances made in the semiconductor industry, such as lithographic printing, it was now possible to design and mass produce thin-film heads that were millionths of an inch in dimension.

On the Memorex 3680, for example, Sordello said there were four thin-film heads on each end of a spindle arm, two to service one side of a platter. These heads made it possible ot record 12.5 million bits of data for every square inch of disk surface.

Gary Kildall asked if there were any mechanical limitations to this technology. Or would you just keep seeing greater and greater recording densities? Sordello said Memorex's own internal plan had projected the limitations. He noted there were three key measurements to consider: (1) the number of tracks per inch along the radius of the disc platter; (2) the number of bits-per-inch along the circumference; and (3) the product of multiplying the first two numbers. Memorex's current media had 800 tracks per inch and 15,000 bits-per-inch, for a total of 12.5 million bits per square inch. Through the use of additional thin-film heads and thin-film media that Memorex planned to debut shortly, Sordello said he expected that areal density would increase by a factor of 16 to 200 million bits per square inch.

Cheifet asked about cost factors involved in producing thin-head technology. Sordello said the recording head consisted of two parts: the larger air-bearing slider and the smaller magnetic transducer (i.e., the "bent nail"). Compared to the older technologies, the cost of the slider was about the same with the thin heads. But the transducer will eventually have a lower cost. Five or six years from now, Sordello said the total product cost would be about 3 to 5 percent less than it was now. At the same time, the thin-head technology performed much better than the older heads.

Cheifet asked if Memorex's thin-film technology only applied to Winchester hard drives or could it also be used in floppy disk drives? Sordello said the technology applied to all forms of magnetic recording. He noted there was currently still some problems regarding wear on the "relatively soft" thin-film materials. But once this problem was resolved, thin-film could be used in floppy disk and even tape drives.

Cheifet then asked Alan Shugart if he saw this technology as a factor in what Seagate was doing. Shugart said not at the present time, because the recording densities required of his products did not justify it. But at some point in the future, he expected that thin-film heads would be used in smaller, more cost-sensitive disk drive products.

Cheifet ended the program by displaying yet another new storage technology, the Drexon LaserCard (see image below) from Drexler Technlogy. This was a small laser-optical storage device that could hold 2 MB of data. Cheifet asked Sordello if he believed that this type of optical storage was the eventual replacement for magnetic storage. Sordello said that once there was real-time, erasable optical media, he believed there would be an "exponential decay" in magnetic recording. Sordello noted that optical recording offered the promise of 2 billion bits per square inch, and that the first non-erasable optical device in development now were in the range of 250 million to 500 million bits per square inch.

# Alan F. Shugart (1930 - 2006)

While he may have come across in this episode as an obstinate fool unable to see past his company's own successful 5.25-inch floppy drives, Alan Shugart really was one of the true pioneers of the field. As his obituary in the *New York Times* detailed, Shugart's career began at IBM in the early 1950s, where he started out repairing punch card machines. In 1955, Shugart was part of the IBM team that developed the first magnetic disk drive, the RAMAC. Eventually, Shugart rose to become IBM's director of engineering and managed the development of a 50 MB disk system that later formed the basis of Sabre, the first online reservation system for airlines.

Shugart and several IBM engineers left the company in 1969 to join, coincidentally enough, Memorex. Three years later, Shugart formed his own company, Shugart Associates, which produced one of the earliest 8-inch floppy disk drives. After Shugart was forced out of the company in 1974, the *Times* said he took a five-year break from the industry before re-emerging to form Seagate in 1979. That same year, Seagate produced the first 5.25-inch floppy drive, as well as a 5 MB hard drive that cost $1,500.

Around the time of his *Chronicles* appearance, Shugart was replaced as president of Seagate by the company's chief operating officer. Shugart returned to power in 1991 and served again as CEO until his retirement in 1998.

## Frank J. Soredello (1937 - 1999)

There's less public information regarding Frank Sordello, who passed away in 1999. According to a *Washington Post* obituary, Sordello spent his career as an engineer. He held 44 patents–38 of which he had obtained before his *Chronicles* appearance–and like Alan Shugart, he started his career at IBM. He left Big Blue to co-found his own disk drive manufacturer, Information Storage Systems, before joining Memorex in the late 1970s as a vice president.

## Notes from the Random Access File

- This episode is available on the Internet Archive and has a broadcast date of May 14, 1984, although my guess the actual recording data was sometime in late 1983.
- The term "Winchester" refers to the IBM 3340 hard disk drive, which was released in 1973. According to an account published by the Computer History Museum, IBM project manager Kenneth Haughton reportedly joked the new drive–which had a 30 MB capacity–was like a .30-30 Winchester rifle cartridge. Winchester then became a generic term for describing early hard disk drives.
- The Memorex 3680 was a behemoth. According to Memorex's product specifications, the assembled drive was nearly 3-feet tall and weighed 625 pounds. As noted by Frank Sordello, the 3680 had a storage capacity of 1.25 GB.
- I could not find the original retail price of the Memorex 3680, but according to the April 22, 1985 issue of *ComputerWorld*, Memorex sold a "double-deck configuration" of the drive–with a whopping 10 GB of storage–for $310,310.00.
- The "bubble memory" displayed by Gary Kildall could be thought of as a forerunner of the modern solid-state drive. According to an article by John C. Dvorak, Bell Labs invented bubble memory in the 1970s. It recorded data on "bubble-like magnetic regions on the surface of a chip." But the fabrication process "never proved to be smooth or cheap enough to compete with other technologies," although it was used by the military for many years.
- Similarly, the RCA Capacitance Electronic Disc (CED) was a complete market bust. To reiterate, the CED was not an optical-style disk like a CD or DVD. It was more like a vinyl record, where a mechanical stylus had to read the data off of the disc. The YouTube

channel Techmoan published a video in 2016 if you’re interested in learning more about the CED.
- The Drexon LaserCard shown by Stewart Cheifet at the end of the episode was a laser-recordable optical memory card introduced in 1981. It was designed to work with a special read/write apparatus developed by SRI International for the Apple II, according to inventor Jerome Drexler. The technology is actually still in use, primarily for government identification cards.

# Chapter 18: Personal COBOL, Forth, and DR Logo

In Part 16, we saw a demonstration of Apple Logo, a computer programming language promoted as an alternative to BASIC. This next episode of *The Computer Chronicles* features another version of Logo–this one developed and sold by co-host Gary Kildall's Digital Research–as well as a broader discussion of the state of computer programming languages around early 1984. The other languages presented in this episode–COBOL, Forth, and Pascal–are still in use today, even if they are not necessarily at the top of the Stack Overflow survey of most popular programming languages.

## Computers Can Speak to Humans–But Not Each Other

Stewart Cheifet opened this episode by posing a simple question to Gary Kildall in French. Since Kildall didn't speak French, he didn't understand the question. Cheifet joked this demonstrated how a human "computer" could not understand a language different than what it was programmed with.

Cheifet asked Kildall to explain the different levels of computer programming languages. Kildall said there was basically three major "levels" of languages: (1) machine language made of up 1s and 0s, which formed the lowest level; (3) system languages like C, which were used to create applications like word processors and spreadsheets; and (3) high-level end-user languages like Fortran, COBOL, and Logo.

Cheifet then narrated our B-roll for the episode, which focused on the problems of having so many different computer languages. There was a demonstration of a "multi-lingual keyboard" that used software to translate English sounds to Japanese text. Cheifet noted this program did for human beings what most computers were incapable of doing with their own languages. Because machine code varied from one microprocessor to another, the same programming language could not be shared by two different computers. For example, Apple BASIC was not the same thing as IBM BASIC. A language customized for one computer would not run on another.

Cheifet said that like their human-language counterparts, computer programming languages differed in structure, syntax, code, and even symbols. But the lack of any standard transferrable code put the programmer at a time-consuming disadvantage: After learning to write a program for one computer, they had to rewrite it for another. And because of a computer's wide range of applications, languages tended to be specialized–e.g., Fortran was used for science and math, COBOL for business, and Logo for education.

The more friendly that a higher-level language was for the user, the more translation it required before it could be executed at the machine level, and thus the slower it ran. In some cases, Cheifet said, the lack of speed at the machine level was a trade-off for quick interaction with the user. Adding a third layer of complexity was the assembly language required to translate higher-level programs into machine code. Conversely, programs written in assembly were faster in execution but more difficult to write. Consequently, software companies were developing more powerful "portable languages" for use in microcomputers, but as long as manufacturers designed exclusivity into their products and languages, users were faced with the same dilemma–machines that could communicate with everyone except each other.

## "Exploding Myths About COBOL"

Paul O'Grady, a co-founder of Micro Focus–the sponsor for this early run of *Computer Chronicles*–and J. David Eisenberg joined Cheifet and Kildall in the studio. Kildall asked O'Grady about Micro Focus' new "Personal COBOL" offering. O'Grady said his company was in the business of "exploding myths about COBOL." He noted that mainframe users held to the myth that COBOL could not run on smaller microcomputers. And microcomputer users held to the myth that they just didn’t like COBOL. O’Grady said the goal of Personal COBOL was to differentiate between the language itself and the environment. In the past, COBOL operated exclusively in the mainframe environment, and that had led to much dissatisfaction.

O’Grady then demonstrated Personal COBOL on an IBM Personal Computer. He described this version of COBOL as a combination of an editor, forms generator, and syntax checker. Specifically, he showed how you could generate a data-entry form in roughly five minutes.

Kildall noted that in traditional COBOL programming, you would work out all of the details that would produce this screen. But in this case, O’Grady made the screen manually by moving a cursor around, effectively the inverse of traditional COBOL. O’Grady said that was correct. With Personal COBOL, the user created the form and the software generated the COBOL code automatically.

Cheifet followed up on O’Grady’s statement that some programmers have said in the past they did not like COBOL. Why was that? O’Grady said COBOL involves a lot of detailed syntax, such as placing a period at the end of each statement, which created problems for programmers. There was also the need to define records at the start before getting into “the guts” of a program. This reflected the type of business applications that COBOL was designed for, such as data-handling and file-manipulation programs. For these kinds of applications, O’Grady said it was necessary to create the records at the start.

Cheifet then turned to Eisenberg, a senior software engineer at Apple, about his work with the Pascal programming language. What did Eisenberg like about Pascal? Eisenberg said one of the nice things about Pascal was its power of expression. In terms of a small amount of on-screen text, the user could get a “lot of power” out of Pascal. For instance, you could express a mathematical algorithm the way you think about it instead of forcing it into some other mold. Much like COBOL’s approach to business applications, Eisenberg said Pascal adopted a similar methodology for more general purpose applications.

Kildall noted that several years ago, Pascal had generated a lot of interest as an all-purpose systems language, but that seemed to be waning. Eisenberg pointed out that the inventor of Pascal, Niklaus Wirth admitted the original goal of the language was to teach people about computer science. Indeed, Wirth was now working on another language, Modula, that was supposed to take over from Pascal as a systems language.

Cheifet returned to O'Grady, asking him what he saw as the use for Personal COBOL. O'Grady said he saw Personal COBOL moving beyond the traditional business applications into more personal uses like diaries and personal filing. He also saw its potential for office automation applications, basically anything that required data handling and file maintenance. Kildall added that COBOL was not really a language meant for end users, but rather for developers who wanted to write and sell programs. O'Grady quipped that as far as he was concerned, none of the programming languages currently available were really meant for end users, including BASIC.

## "When You Discuss Languages, It Almost Becomes a Religious Argument."

For the final segment, Elizabeth Rather joined Cheifet, Kildall, and Eisenberg. Rather was president of Forth, Inc., which produced the Forth programming language. Cheifet asked Rather what made Forth unique. She said there were two things. First, it was the only programming language designed from "first principles" to work on a small computer doing interactive software development for real-time applications. Second, Forth worked on all three levels of programming that Kildall discussed during the introduction. It could work at the machine code, systems programming, and higher user levels.

Rather demonstrated Forth on an IBM Personal Computer. She noted that the operating system used to run the demo was itself written in Forth. She demonstrated a simple graphics application that generated and displayed the Forth corporate logo and performed a series of area fills.

There was then an exchange between Kildall and Rather over the combination of assembly language and Forth in the code used to produce this program. Rather clarified it was a mix of the two. With Forth, you could put in as much assembly as needed, either to control the hardware directly or to make things run faster.

For the final demo, Kildall took control of the IBM PC himself to show his company's DR Logo. While Kildall got the program up and running, Cheifet asked Eisenberg if there would always be a number of different computer programming languages? Or would they merge into a smaller number? Eisenberg said while it would be nice to have an "ideal language," he didn't think that would ever happen. He reiterated how different languages were designed for different purposes. He also noted, "When you discuss languages, it almost becomes a religious argument." But at the end of the day, a programming language was simply a tool. It would be like arguing over whether a hammer was better than a screwdriver–tell me what you wanted to do, and he would tell you which was better for that job.

With that, Kildall got DR Logo running. He noted it was an interpreted language that gave the user immediate feedback to the work they were doing. For example, if you typed `FD 40` in the turtle graphics portion of Logo, the on-screen turtle moved forward by 40 steps. In contrast to this approach, other programming languages like Fortran and Pascal were compiled, meaning they did not give this type of immediate feedback. The programmer had to go through the process of editing and compiling and often a few additional steps, which required a lot of abstraction.

Kildall also demonstrated an application written in DR Logo, a simple blackjack game designed by a 13-year-old friend of his son. Kildall said the game was written in about 3 or 4 days and showed how you could build a program without a great deal of effort that was still interactive and graphical in nature. Eisenberg added that one of the nice things about languages like Logo was that the user could define their own words, which made the final program read closer to English.

Cheifet noted that most new computer users were exposed to BASIC as their first language. How was Logo better than BASIC? Kildall explained that BASIC was essentially derived from Fortran, the first high-level programming language, and that meant BASIC also inherited a lot of Fortran’s problems. It required a step-by-step breakdown of a problem into simple steps. That was not always the best approach for building a program. Logo, in contrast, came out of work done at MIT where the developers thought about the “problem of problem solving itself” and created a language that would suit that purpose.

Kildall said that BASIC was an easy programming language to learn but it had limitations as soon as you started to do anything complicated. Rather added that BASIC was designed for use by college undergraduate students performing simple tasks. But newer languages like Forth and Logo had the capacity to grow and learn new things, not unlike the process of teaching a small child. Kildall said Logo also provided something of a “bait and switch.” It got children interested through the turtle graphics, but behind it was a form of Lisp, another programming language that had been used for years in work on artificial intelligence and list processing. Forth and Lisp also shared some foundations.

Finally, Cheifet asked if there were any new programming languages on the horizon. Eisenberg remarked that at last count, there were about 700 or 800 known languages. But many were designed to perform a specific task or were offshoots of one of the main languages. But as long as people had ides for things they wanted to do, they would continue to invent new languages. Kildall added that once you learned one programming language, it was not very difficult to pick up other languages. The threshold was to learn that first language.

## Visual Cobol and SwiftForth Carry On the Legacy

I mentioned the annual Stack Overflow survey in the introduction. This polls developers on, among other subjects, their primary programming, scripting, and markup languages. You won’t find any of the languages discussed in this *Chronicles* episode on that list. That’s not to say, however, that COBOL, Forth, Pascal, and even Logo are no longer in use.

To the contrary, more than 60 years after the language was first deployed, Micro Focus continues to actively develop and market COBOL. The company's Visual Cobol 7.0 can presumably trace its lineage back to the Personal COBOL demonstrated on *Chronicles*. Only today's COBOL now works with modern editors like Microsoft's Visual Studio Code and can compile applications into more commonly used programming languages like Java.

Forth, Inc., also continues to produce its namesake programming language in more modern forms. In addition to consulting services, Forth offers SwiftForth, an integrated development environment compatible with Windows, Linux, and macOS. There are other implementations of Forth developed outside of Forth, Inc., which are commonly used in embedded computer systems.

## "Computers Were Built to Make Money, Not Minds"

As discussed in my prior post on the Apple Logo episode, its famous turtle graphics lives on in the present-day Python programming language. Gary Kildall would no doubt be pleased. This episode showed Kildall's passion for Logo as an introductory programming language for end users, and students in particular.

In an article for the June 1983 issue of *Byte Magazine*, Kildall and co-author David Thornburg argued that DR Logo "sets a new benchmark by which to measure the properties of useful computer languages." They noted that DR Logo, which retailed for about $100, ran on 16-bit machines like the IBM Personal Computer and thus offered a significant advantage in terms of memory over the 8-bit Apple computers running Logo. At the same time, DR Logo was a "superset of Apple Logo," and programs written for the former would work on the latter.

Aside from introducing kids to turtle graphics, Kildall and Thornburg said DR Logo had many other potential uses in educational applications:

> In the physical sciences, Logo can be used to construct *microworlds* in which bodies obey different natural laws, such as gravitation. By

> exploring these artificial microworlds, children can develop better intuitions about the properties of their own corner of the universe.
>
> Given Logo's powerful list-processing capability, one would expect it to be of value in the language arts as well. To pick one simple example, suppose a child created several lists called `nouns`, `verbs`, `adjectives`, `articles`, etc., and assigned appropriate words to each list. The word order in each list can be randomized with the `shuffle` command, and a random sentence can be constructed by assembling words from each list in a syntactically correct valid order. Legitimate nonsense sentences can be automatically generated in this fashion (e.g., No yellow toad smells tall people.) while bringing the child to look at and solve the structure of English.

Later, Kildall conceded that Logo had failed to displace BASIC as the language of choice for introducing students to programming. In an unpublished manuscript for his never-completed autobiography, Kildall said he "felt the kids using BASIC on the Apple II and IBM's new PC were being taught archaic mind tools to solve problems." But while Logo "became popular among a largish cult of teachers that were computer literate":

> [I]n reality, most teachers found themselves racing to catch up with their brightest students and found solace in using BASIC.
>
> This is not a comment about inadequacies in our educational system. It is a comment about the times. I expected too much of educators. I expected them to understand, in a sense, the sugar-coated concepts of LISP used in [artificial intelligence] that were embodied in the Logo language.
>
> It was then I learned that computers were built to make money, not minds.

## Notes from the Random Access File

- This episode is available at the Internet Archive and has a recording date of February 9, 1984.
- Paul O'Grady, who was described as an executive vice president of Micro Focus in this episode, eventually became CEO of the company, a role he served in until the mid-1990s.
- Elizabeth Rather was the second person to actually program in Forth. She learned Forth from its creator, Chuck Moore, who wrote in a 1991 paper that he initially hired Rather to "provide on-site support" for a computer-operated telescope using Forth in Tuscon, Arizona. Rather went on to serve as president of Forth, Inc., until her retirement in 2006.
- J. David Eisenberg worked at Apple from 1979 to 1985. He's worked since then as an independent software developer and a computer science teacher at Evergreen Valley College in San Jose, California. You can watch some of his lectures on his YouTube channel.
- It was only touched on in the episode, but Eisenberg worked on Apple Pascal, which was produced between 1979 and 1985 for the Apple II and Apple III.
- According to a June 1982 company newsletter, Digital Research also had an agreement to sell COBOL compilers made by Micro Focus.
- If you want to hear more of Gary Kildall's thoughts on DR Logo, he gave a lecture on the subject to the Boston Computer Society in March 1983, which has been preserved by the Computer History Museum on YouTube.
- In the previous post on Apple Logo, I mentioned Sean Tibor and Kelly Paredes' podcast, *Teaching Python*. They were kind enough to discuss my post in their most recent episode.

# Chapter 19: The HP 2700 and the Apple Macintosh

When Apple released the Macintosh–later known as the Macintosh 128K–in January 1984, its main selling point was the graphical user interface (GUI). Although the original Macintosh operating system's GUI was largely based on what Apple deployed on the Lisa a year earlier, the company believed the new machine's lower price point would make the interface more accessible to a larger audience.

Of course, the Macintosh was not exactly cheap, even by 1984 personal computer standards. As Gregg Williams noted in the May 1984 issue of *Byte* magazine, a "usable" Macintosh system, including a second disk drive and basic office software, would cost $3,879. This was nearly $1,000 more than the price Apple quoted for the same setup a few months earlier, Williams noted, and it undercut Steve Jobs' marketing claims that the Macintosh would be "something really inexpensive so that everyone can afford it"

Williams also pointed out that many "traditional computer users" were resistant to the Macintosh interface design, complaining that the "mouse, the windows, and the desktop metaphor [were] silly, useless frills," while others were "outraged at the lack of color or graphics." The 128 kilobytes of memory was also far too small for many users' needs, which prompted Apple to release a 512 KB Macintosh just a few months later.

## Adapting Computer Design to Human Thought and Movement

Despite these criticisms, when the Macintosh made its in-studio debut on *The Computer Chronicles* in March 1984, there was still great excitement over Apple's new machine and its GUI. Herbert Lechner of SRI International joined Stewart Cheifet for this episode, where the formal subject was computer ergonomics or “human factors engineering.” The episode began with Cheifet showing off Apple’s first machine–the Apple I from 1976–and noted “you had to supply your own lumber” to build the machine’s case and it was not very user-friendly.

Cheifet then asked Lechner about a recent study from SRI about ergonomics. Lechner said that study identified a number of areas for improving the man-to-machine interface. This ranged from hardware-specific items like the design of the keyboard and the height of chairs to issues affecting the general workspace, such as heat, noise, and lighting. Ergonomics was also a personnel issue, Lechner said, in that it was important for businesses to ensure that the people who were using computers and related equipment were part of the selection and decision-making process.

Cheifet’s customary B-roll narration then provided a brief history of ergonomic design, starting with sophisticated aircraft instruments developed in the 1940s and 1950s. Cheifet said these instruments created human interface problems that were entirely new to industrial designers. Mechanical devices then gave way to electro-mechanical and later electronically controlled mechanisms, which generated more and more information for the operator. In its earliest days, ergonomics was therefore concerned with the safety of the people who used these devices in applications where error could cause injury or death. But as the use of ergonomic design became more common, it quickly spread to other working environments with the goal of increasing productivity and reducing fatigue and stress.

Cheifet noted that among computer users, some major health concerns started to surface in the 1970s regarding pain, stress, and especially eye fatigue. After all, sitting in front of a video display was a radical change for most workers. It led to a study of how the eye responded to stimuli from a cathode-ray tube (CRT) display. Experiments measured a user’s eye irritation arising from screen glare, which helped to chart the benefits of using anti-glare displays. Ultimately, Cheifet said, the goal of computer ergonomics was to adapt machine design to human thought and movement.

## “We Have Adjustable Chairs, But I Wonder If We Have Adjustable Software”

In the studio, Wanda Smith and Karen Kessel joined Lechner and Cheifet. Lechner opened by noting that not everyone was familiar with the term “ergonomics.” Some people thought it had to do exclusively with the interaction between people and computers. But there were also software and physical interactions to consider. He asked Kessel, an ergonomics consultant with the Koffler Group and editor of its *Office Systems Ergonomics Report*, to explain what she believed ergonomics meant. Kessel replied it was basically the way that Lechner described it. Ergonomics had to consider the entire workstation design, including the terminal and software. The goal was to look at the user and their task. What were the user’s needs? How was the user constrained by their task? What was the user’s memory capabilities? In answering these questions, ergonomic design was supposed to try and tailor the environment so that it best suit the user.

Cheifet turned to Smith, the manager of human factors engineering for Hewlett-Packard, and pointed out that a computer or terminal keyboard was largely based on a typewriter. Did that create a problem for users transitioning to the computer? Smith said there was some “transfer of training” problems when moving some users from the typewriter to the computer, especially if they had never used the latter before. For example, some users kept hitting return or a button that accidentally erased their data. But in her experience, she found such transition problems were overcome “very quickly.”

Lechner pointed out that not long ago, there was quite a bit of concern expressed by users of video display terminals (VDTs) about their safety and physical comfort. Since that time, there have been some changes made to terminal design, such as detachable keyboards. He asked Smith to elaborate on these changes. Smith said that when users started expressing these concerns, there was both a number of product design changes, as well as legislation in Europe mandating such changes, which made VDTs accommodate the user’s needs better and in turn made the entire system more adaptable and flexible.

Cheifet asked Kessel about other ergonomic issues that users faced on the hardware side. Kessel replied that lighting, on-screen character quality, and other aspects of the user’s physical workstation all played a role. For instance, if a worker used a terminal but also needed information from a telephone, it was important to make sure that telephone was close to their workstation. With respect to VDTs, the main concerns were the quality of the screen and the keyboard.

Cheifet followed up by asking about the software side of ergonomics, which he said appeared to be a “more subtle” problem. Kessel said it was also a more difficult issue, as it required an assessment of the cognitive capabilities of the user. The software had to anticipate what the user was thinking and figure out what steps they needed to take to achieve a certain goal. Lechner added the software also needed to be flexible, as not all users were the same. He quipped, “We have adjustable chairs, but I wonder if we have adjustable software.”

Smith then demonstrated one of her company’s products, the HP 2700 terminal. (Cheifet referred to this machine as the “HP Orion.”) The demo contained a PERT Chart, a common tool used in project management. Using the HP 2700’s graphical capabilities–the terminal could display 4,029 different colors–the information in the chart could be modified to add accent colors to highlight certain information on an otherwise black-and-white chart.

Smith also demonstrated the use of multi-color design in a word processing application. She pointed out the user could design something like a form letter with different colored sections to highlight fixed areas versus those items that needed to be changed on a daily basis. Smith said H-P was starting to use multi-color design to study the visual and cognitive responses of users, which as Kessel noted was a more difficult problem than hardware design.

Cheifet also asked about the thumb-wheel included on the HP 2700’s keyboard, which was kind of a mouse. Smith said it was a mouse. But the advantage of having a thumb-wheel designed into the keyboard was that it did not require additional space on the tabletop like a traditional mouse or interface device.

## Icons Make Macintosh Design “Analogous to Life”

The final segment was devoted to the Macintosh. Susan Kare, one of the lead designers for the Macintosh user interface, and Jerry Manock, the product manager, joined Cheifet and Lechner with their new creation. Cheifet asked how the Macintosh addressed some of the ergonomic problems discussed throughout the episode. Manock said the goal was for Macintosh to be used by the “99th percentile” of users, including men, women, and children. This dictated certain base criteria for the Macintosh, including the keyboard, the angle of the screen, and the placement of the monitor above the disk drive slot.

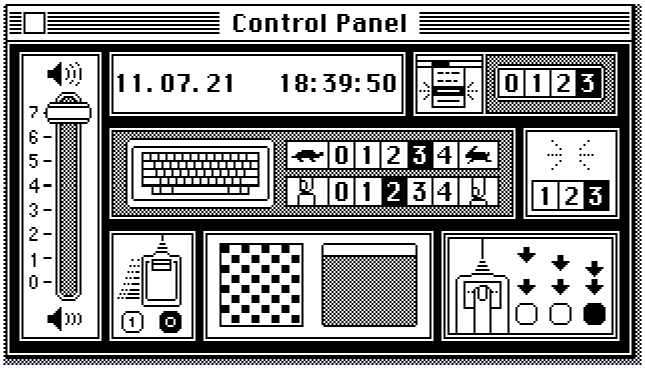
Manock said “transportability” was also a major goal. He pointed to the Macintosh’s light, detachable keyboard and the handle on the top of the computer. The Macintosh also had a “unique” disk drive that moved away from “soft, easily damaged” 5.25-inch diskettes to the “hard, plastic shell” 3.5-inch diskettes. The 3.5-inch design not only protected the media inside, it made it easier for users to carry diskettes in their pocket and ultimately made the drive itself more reliable in any climate.

Cheifet next asked about the Macintosh’s one-button mouse, which had also been a key part of the Lisa’s design. Manock said there were a number of studies showing the mouse was a “very efficient cursor pointing device.” And while early computer mice, including those made by Apple, relied on 2 or 3 buttons, “We worked very hard to get just the button to be a single button without having to have the person remembering right or left.” The user just had to click one button.

Lechner then asked Kare about her work as an icon designer. What did she learn from her experience on the Lisa regarding the use of icons and how they fit into ergonomic design? Kare said the goal was to have someone who had never used a Lisa, a Macintosh, or really any computer before learn how to operate the machine in about 20 minutes. A lot of this came from the fact you could explain what an icon meant and have the person remember it. The Lisa provided a jumping off point in this regard and Kare made some refinements and additions for the Macintosh.

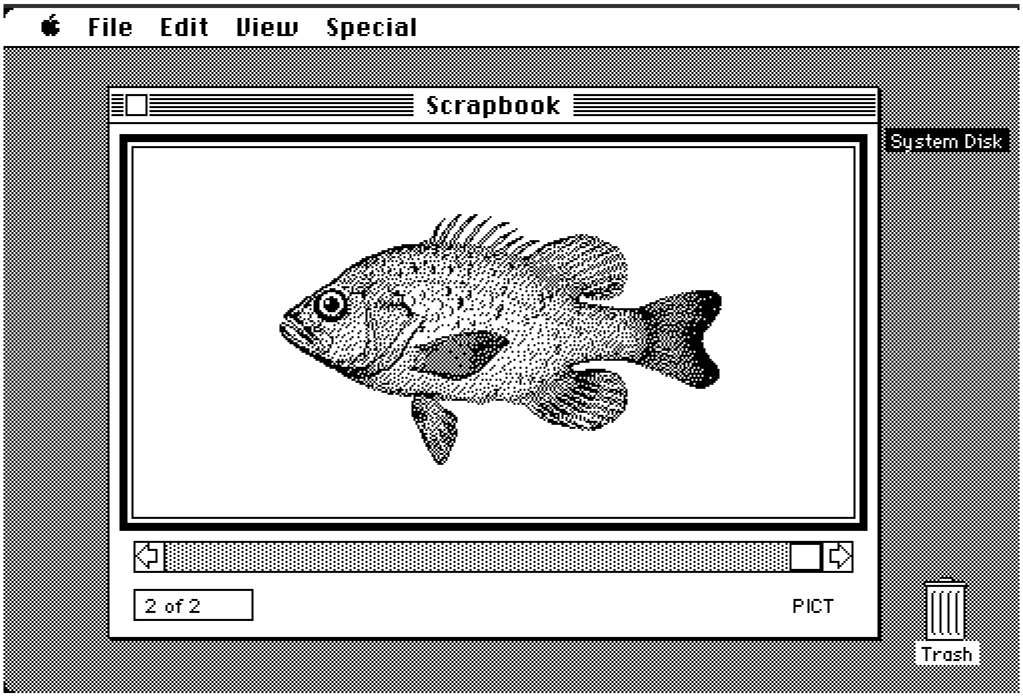
Kare then provided a demonstration of the Macintosh itself. She made note of the extensive use of icons, such as the “disk with a question mark” at startup that prompted the user to enter a diskette. There was also the “content Mac” when things were operating normally. The Macintosh “desktop” allowed users to highlight specific icons with the mouse, causing the icon to invert. Kare said the “hidden menus” activated from the top bar on the screen operated like “window shades.” Graphical windows displayed the contents of individual diskettes and storage folders. All of these design choices were, as Kare put it, “analogous to life.”

Cheifet asked for a demonstration of the Macintosh Control Panel. Kare described this panel as a “dashboard” that let the user fine-tune the system in a number of ways to suit their comfort level or personal preferences. The computer itself would work fine no matter the settings. For example, there was a volume control slider on the left side of the Control Panel that altered the speaker sound. Kare said if a person was at home by themselves and listening to the stereo, they could raise the volume to ensure they always heard alert beeps from the system. Conversely, if the user was working on a Macintosh in the library, they could set the machine to make no noise and just flash to get their attention. Other settings adjusted things like the keyboard repeat rate and the time between registering double-clicks with the mouse.



The Control Panel from the Macintosh System 1.0.

Kare also demonstrated a few of the Macintosh's built-in applications, including Notepad and Scrapbook. Notepad allowed the user to write quick notes and "flip" between up to 8 separate pages. Scrapbook–see the image below–could store up to 256 pictures or messages in memory, which could then be pasted into other documents. Kare also briefly demonstrated MacWrite 1.0, a WYSIWYG word processor, that included features like a ruler to set margins and buttons to make formatting changes. Cheifet noted the user could also pull pictures from Scrapbook and paste them into a MacWrite document.



Monochrome image of a fish included with the Scrapbook application on Macintosh System 1.0.

## The First "Electronic University" Announces Plans to Offer Degrees

Stewart Cheifet presented this week's "Random Access" segment, which was recorded in March 1984.

- TeleLearning Systems announced it would create an "electronic university," the first nationwide computer-based college offering credit courses and degrees. TeleLearning said it would work with seven major universities throughout the United States.

- Four Japanese manufacturers announced the production of the first 1 MB ROM chip.
- Scientists at the Lawrence Livermore Laboratory in California unveiled the world's fastest supercomputer, the Cray X-MP, which could perform 200 million calculations per second. Cheifet said the X-MP would be used primarily for research on President Ronald Reagan's Strategic Defense Initiative (i.e., the "Star Wars program.")
- Texas Instruments said it would start selling a voice interface for its professional line of computers. The interface could understand up to 50 spoken commands. Cheifet said it would be available in April 1984 and retail for $2,600.
- Kaypro announced a new IBM-compatible notebook-sized portable computer that would be priced at the "high end" relative to other Kaypro models.
- Meanwhile, Cheifet said IBM was taking aim at the various clones of its popular Personal Computer. The recent introduction of IBM's own Portable PC 5155 was a step in that direction. Cheifet said there were also rumors that IBM planned to dramatically increase production, drop prices, and develop a new computer that was harder to copy. Yet despite increasing competition, IBM was still the dominant player in the personal computer market, with a $6 billion *increase* in sales during 1983.
- Harvard Business School announced it would require all of its students to have an IBM PC starting in the fall. Harvard's undergraduate college, however, decided to go with the Apple Macintosh as the required machine for its students.
- A group of shareholders filed a class action complaint against Apple over alleged misrepresentations regarding the Lisa. The lawsuit also accused Apple executives of selling off nearly 2 million shares of stock before revealing a lower-than-expected earnings report.
- Paul Schindler reviewed the game *Archon: The Light and the Dark* from Electronic Arts, which retailed for $40. Schindler enjoyed the game, which he referred to as an "addictive cross between *Dungeons & Dragons* and chess."

- Cheifet noted that we soon might be referring to software stores that used to sell books. Two major bookstore chains were now carrying computer software. And a major book publisher was now distributing software to bookstores.
- Finally, Cheifet briefly discussed a new self-evaluation program for Apple II computers, *Unbecoming a Hero*, which used a series of questions to determine the user's values and offer them a role model that met those values. The software sold for $30.

## HP's "Most Advanced Terminal" Fizzled in the Market

As mentioned in the introduction, one of the early criticisms of the Macintosh was its lack of color graphics. The other product demonstrated in today's episode, the HP 2700, certainly did not lack in that department. According to a profile on the HP Computer Museum, the 2700 was "the most advanced terminal built by HP in its first 25 years in the computer industry." The terminal came with a graphics tablet and could produce bitmap graphics with thousands of colors at a 512x390 resolution.

But it did not come cheap. The base configuration cost $24,000, and "top-of-the range models" sold for around $28,000. As a result, the product had difficulty selling and was already considered obsolete by HP at the time of its *Chronicles* appearance in 1984.

## Susan Kare: Mother of Modern Computer Iconography (and Clarus the Dogcow)

The Macintosh, of course, lives on in name today. The actual 128K model seen in this episode was quickly superseded by the upgraded 512K unit. But the basic iconography and design produced by Susan Kare would continue to define the Mac aesthetic for decades.

Oddly enough, Kare had little to do with computers before joining Apple in 1983. According to a 2019 profile for *Smithsonian Magazine* by David

Kindy, Kare was working as a sculptor when an old friend–Apple designer Andy Hertzfeld–asked her to apply for a job “creating graphics and typefaces for the new personal computer Apple was planning to release in 1984.”

Kare got the job and proceeded to design a number of original typefaces for the Macintosh. Stephen Hackett, the unofficial Mac Archivist of the United States, cataloged Kare’s typeface designs in a post for his 512 Pixels blog. Hackett noted this included Chicago, a sans-serif font that “was the default system font up to [Macintosh] System 7.6 and later appear[ed] on iPods, as it renders well on black and white displays.”

Hackett also pointed to Kare’s work on the Cairo font, which he described as “the original dingbat font,” i.e., a typeface composed of graphical symbols that were not traditional letters or punctuation characters. Of particular note, Cairo’s replacement for the traditional “Z” character was a “small creature named ‘Clarus’,” which another Apple employee later dubbed “The Dogcow.”

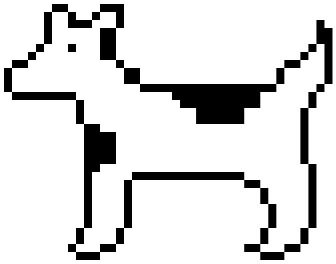


Image of Clarus the Dogcow from the original Macintosh Cairo font.

But as we learned in the *Chronicles* episode, Kare’s contributions went well beyond fonts into the actual system icons. Kare later recounted the process to *Smithsonian*:

> [Andy] Hertzfeld created an icon editor on the [Macintosh] prototype, from which Kare could create electronic versions of each icon with a mouse.
>
> “At the time, the ability to design on the screen seemed amazing,” she says. “It was possible to undo, and iterate, and design an icon or

> letterform while simultaneously seeing it enlarged and at 100 percent. It was exciting, and felt like a magical leap forward."
>
> Kare goes on to say, "Decades later, where working with sophisticated paint tools and multiple levels of undo is commonplace, it's easy to forget how enjoyable it was to experience the most basic digital tools."

Kare remained at Apple until 1986, when she left to serve as creative director at Steve Jobs' NeXT, Inc. In 1989, she started her own design firm. As *Smithsonian* noted, Kare continued to create a number of memorable designs in the technology space, including "the playing cards for Microsoft's Windows 3.0 Solitaire game in 1990 and the virtual gift icons she developed for Facebook in 2007." In 2015, Kare joined Pinterest as its creative director.

## Notes from the Random Access File

- This episode is available at the Internet Archive and has a recording date of March 29, 1984, although it's worth noting the on-screen demo of the Macintosh Control Panel displayed a date of March 14, 1984, at 3:56 p.m.
- In recent years, former Hewlett-Packard manager Wanda Smith has made a name for herself in the equestrian field. She is the president of the California Equestrian Park & Event Center (CEPEC), an organization that provides "recreational, educational, competitive, and equine assisted therapy venues for youth and adults" in Sonoma County. Smith has also worked as a substitute teacher in recent years and authored a book, *Horses of the Wine Country*.
- Jerry Manock lives in Vermont. According to a 2012 profile in the Veremont publication *Seven Days* by Paula Routly, Steve Jobs paid Manock $1,800 in 1977 to help Steve Wozniak design the original Apple II. Manock said he asked the "dubious" looking Jobs for the money upfront. Manock continued to work with Apple through the launch of the Macintosh in 1984. Before and after his Apple tenure, Manock worked as an independent design engineering consultant and he taught classes at the University of Vermont for more than 20 years.

- Unfortunately, I could not locate any definitive information regarding the whereabouts of Karen Kessel. The Koffler Group, the consulting firm she worked for, dissolved in 1991. The company was founded by Richard P. Koffler, who later ran for Congress in California as a Libertarian Party candidate.
- Jerry Manock touted the Macintosh's "portability" quite a bit in this episode. Just to be clear, the Macintosh weighed just under 20 pounds when you factored in the keyboard and mouse. Of course, that was about half of what an IBM PC with a monochrome display weighed.
- TeleLearning Systems' "Electronic University Network" was started by Ron Gordon, a former president of Atari. In a January 2018 article, Cait Etherington delved into what became of the project. She noted the tuition was "relatively affordable, ranging from $12 for a seminar up to $295 for credit-based university courses degree." By September 1985, there were about 15,000 students attending 1,700 different classes. But there were a "few bugs," such as students having to wait days for messages to reach their on-line instructors.
- The IBM-compatible notebook teased by Kaypro may have been the Kapypro 2000 laptop, which was released in 1985 and retailed for $1,995.
- *Archon: The Light and the Dark* was one of the first games ever published by Electronic Arts, which was profiled in Part 15 of this series.

# Chapter 20: Expert-Ease and the KEE System

In Part 14 of this series, the *Computer Chronicles* first discussed the subject of "expert systems." This referred to computer knowledge bases that purported to replicate a human's expertise in a particular field. This next *Chronicles* episode revisits the idea of expert systems as part of a broader discussion of artificial intelligence.

## The Link Between the Mechanical World and Abstract Concepts

Herbert Lechner is back as Stewart Cheifet's co-host for this episode. The program opens with Cheifet demonstrating a pocket chess computer–I could not make out the model, but I'm guessing it's from Radio Shack. Cheifet noted that chess was considered the "ultimate game of skill" and required a certain amount of mental agility or "intelligence." And yet, this simple computer was capable of applying a certain kind of intelligence by analyzing board positions and selecting moves.

Cheifet then pointed out that for some people, artificial intelligence (AI) suggested an attempt to duplicate the way that a human brain worked. Was that what AI was really about? Lechner said not in terms of modern AI research. He said early AI research did look into duplicating human thought processes. But the current focus was on duplicating the end result of intelligence as opposed to the process. For instance, this meant developing computers that acted as experts–i.e., expert systems–and machines that could communicate with people in human language.

The normal B-roll was preceded by clips of an interview with Dr. Edward Feigenbaum, a professor of computer science at Stanford University. Feigenbaum explained that computers were general symbol processing devices capable of manipulating many kinds of symbols, of which numbers were one important class. But computers were also much more general than that. He noted that researchers have understood the “generality of computation” since the work of Alan Turing in the 1930s. And Feigenbaum said he’d traced these ideas back as far to “intuitions” expressed by Charles Babbage to Ada Lovelace in 1842. Lovelace said that Bababbe’s proposed analytical engine provided a link between the mechanical world and the most abstract of concepts. Feigenbaum said this was now referred to as the *physical symbol system hypothesis*, which was the basis for present-day work on AI.

In the science of artificial intelligence, Feigenbaum said, researchers used computers to process symbolic knowledge using logical inference methods. In other words, the goal was inference and not calculation in the traditional sense–or knowledge as opposed to numbers.

This led into Cheifet’s narration. He said that one current application of inferential knowledge was in the field of expert systems. The B-roll showed an example of an expert system, an “oil exploration advisor” developed to assist drilling rigs in remote areas. Cheifet explained that the program behaved much like a qualified specialist: It asked questions of the user and then gave advice on how to avoid or correct accidents that were common in drilling operations. If the user was confused by a question, it could ask the program to explain by typing, “WHY?” The “advisor” would then respond with the specific source for its question and explain its reasoning up to that point. Cheifet said this type of symbolic processing had found applications in a broad range of fields from medicine to robotics.

At the same time, Cheifet said, the most difficult area for AI to master was natural language–a talent that resulted in very friendly computers but also required enormous processing capacity. The ambiguities of syntax and context had restricted present AI systems to very limited applications. For example, a simple question about geography could lead to multiple interpretations depending on the phrasing. A program had to determine the question’s most likely meaning after conducting an exhaustive deconstruction of the sentence, and even then it could reject an unusual phrasing.

Cheifet concluded by noting that in a parallel development, there were other applications available that gave persons without the ability for normal speech the ability to adapt or construct alternative symbolic languages–a remarkable example using a computer program's "intelligence" to help interpret human intelligence.

## Copying the Knowledge of "World-Class Experts"

Jeffrey Perrone and Nils Nilsson joined Cheifet and Lechner in the studio. Nilsson, another computer science professor at Stanford, tried to define the current scope of AI. He viewed AI as mainly concerned with putting knowledge into computer programs, so that those programs could then solve certain problems for humans. And that knowledge had to be represented in a particular sort of way.

Lechner noted that it was not just that AI programs were smarter, but that there were also different techniques used relative to ordinary programs. Nilsson agreed there was a body of techniques used with AI that differed from the rest of computer science.

Cheifet then turned to Perrone, a consultant who marketed expert systems, and asked him to explain how the type of AI research performed by Nilsson was applied in practice. Perrone said that AI-based expert systems could now be used anyplace where "specialized knowledge" was used routinely to reach decisions, troubleshooting strategies, diagnoses, and so forth. He noted that we'd reached a watershed where it was no longer very expensive or difficult for individuals with no technical background to build and use such expert systems.

Lechner then asked Perrone to demonstrate an expert system called *Expert-Ease*. Perrone explained that *Expert-Ease* was a tool for generating expert systems applications in virtually any area. It ran on microcomputers like the IBM Personal Computer. And while *Expert-Ease* applied some of the basic AI concepts that Nilsson discussed, the software also had its own unique approach. For example, instead of requiring an explicit statement of rules to build an expert system, *Expert-Ease* only needed examples of previous decisions, tasks, or diagnoses.

The actual demo involved an expert system that Perrone previously developed with *Expert-Ease* with Dr. Hillary Don, an anesthesiologist. The purpose of this particular system was to help diagnose patients with potential airway and breathing problems. The program presented a series of questions to the user, such as:

- When was the onset of the problem?
- What is the quality of the stridor (a rasping sound made when one's breathing)?
- Is the patient drowsy?
- Are there any predisposing factors to developing stridor?

Based on the answers provided, *Expert-Ease* offered a potential diagnosis. Lechner asked who would use a system like this. Perrone said either a doctor or a physician's assistant could use the system to screen patients "for further fine tuning" by a human with more specific expertise.

Perrone also demonstrated a more "trivial example" of the *Expert-Ease* software to illustrate how its underlying matrix worked. This example helped a user decide what to do on a Sunday afternoon based on two inputs–what was the weather like and did the user have their family with them. Perrone explained these questions or "attributes" were used to build the expert models. (Basically, it's a glorified spreadsheet.) For any particular problem, the user could define up to 31 attributes and 255 values per attribute. So it was possible to build very large models with *Expert-Ease*, subject only to the limitations of the host computer's disk storage. After sketching out the attributes, the user then went to an "example screen" to enter examples of previous decisions.

Cheifet asked if it was really possible to seriously approach this kind of problem solving using a personal computer? Perrone said yes. He noted that he'd recently spoken with the *Whole Earth Software Review* about implementing an expert system to recommend software to its readers.

Cheifet turned back to Nilsson and asked him what was the point of all this? Why would we develop something like *Expert-Ease*? Nilsson said there were a lot of uses for this type of system. The present systems admittedly had some "brittle" features that limited their utility at the moment. But over the next 10-to-20 years, he believed expert systems would be generally used in a wide range of settings. The goal was for these systems to contain the knowledge of “world-class” experts. In many fields there might only be 5 or 6 such people. An expert system could help spread the knowledge of that small group in a way that could be easily copied.

## Programs That Can Build Other Programs

The final segment began with Cheifet talking to Tom Kehler of IntelliGenetics, an AI company co-founded by Edward Feigenbaum. Kehler demonstrated the *Knowledge Engineering Environment*, or KEE system, which was an example of a more complex expert system than *Expert-Ease*. Kehler explained the KEE system used a graphical environment to connect underlying knowledge bases. This allowed the user to build models by “just fooling around with” various meters, valves, and objects.

The particular system Kehler demonstrated was designed for a nuclear reactor. The user could take various actions through the graphical interface. Kehler said those actions would then affect a knowledge base that would apply heuristics and decision-making procedures. For instance, the system could test the hypothesis of a particular nuclear accident scenario.

Cheifet clarified that the user could interrogate the system and ask it what course of logic it was following in recommending a specific course of action. Kehler said that was correct. For instance, if the system asked whether a steam-level generator was decreasing, a nuclear technician could ask the system to explain why it was asking that particular question.

Cheifet pointed out the KEE system ran on the Lisp programming language. Why was that the right language for an AI application? Perrone said Lisp was important because you needed very powerful symbolic manipulating capabilities. After all, decision-making was primarily a symbolic activity.

With that, Cheifet rejoined Lechner and Nils Nilsson, along with Dr. John McCarthy, yet another Stanford computer science professor who was also the inventor of Lisp. Lechner asked McCarthy why he invented Lisp. McCarthy replied it was primarily for use in AI work such as the KEE system. Lechner followed up, asking what were the characteristics that made Lisp better for AI than other languages. McCarthy explained that Lisp programs were written in the same format as its data. That made it easy to build programs that could produce other programs. Lechner said it also made it possible to deal with facts and logic more so than numbers. McCarthy said that was correct.

Lechner said when it came to these newer “smart machines,” the question on everyone’s mind was, “How smart can machines become?” In other words, what were the limits of artificial intelligence? McCarthy said he saw no limit short of human intelligence. With faster machines, a computer could do in a short time what a human did over a long time.

Cheifet asked Nilsson for his thoughts on this. Nilsson said there were three problems AI researchers currently faced. The first was deciding what knowledge should be represented in a program. The second was how to represent knowledge in the program itself. Finally, how would that knowledge be used? He noted that while there had already been rather successful activities in all three areas, there were still other things that needed to be done.

Cheifet asked if there was any frustration in the field of AI given that there was a lot of hope, yet it was harder that one might think to develop these concepts into a practical application. McCarthy noted there was already a certain level of practical technology in AI and that people were making applications. On the other hand, he made note of Nilsson’s earlier point that AI programs could now contain the knowledge of world-class experts. McCarthy said that was true, provided the knowledge fit into a format that the present software allowed, i.e., “If this is true, then do that.” But more general kinds of knowledge that could be used in vaguer ways were harder to program in this manner.

Cheifet recalled Jeffrey Perrone’s earlier demo of an expert system used to make medical diagnoses. He noted that there were several such systems available, yet there were problems using them. What were those problems? McCarthy said he was not familiar with those specific systems, but he offered a hypothetical example. Say you programmed a system to define a container as “sterile” if all the bacteria were dead. This knowledge could not be used, however, in a special way to determine if the container was sterile by examining each of the bacteria. It was useful only as general knowledge, i.e., if you heat the container enough, the bacteria will all be killed. Or if you leave bacteria in food, it will spoil.

Cheifet asked where the biggest potential market was for AI applications. Nilsson said the expert systems discussed today represented a rather larger market and that companies would continue to develop “less brittle” systems over the next 5 to 15 years. Another application would be computer programs that could converse with humans in English or other ordinary languages. That would make computers more accessible to a much wider variety of people.

Cheifet asked Nilsson to explain what he meant by “brittle.” Nilsson said it had to do with being able to reason about the context in which a particular discussion or conversation with an expert system was taking place. If there was information that was needed to reach a certain decision, and it was there explicitly, then the system could come up with a reasonable answer. But if the system needed “common sense”–i.e., the type of knowledge most humans had by the time they were 10 or 15–then the system would often fail. Cheifet quipped that ironically, common sense was the most difficult thing for a computer to learn. Nilsson retorted that few people were willing to pay to put common sense into a computer, whereas it was commercially important to put something like knowledge about how to operate a nuclear reactor into a computer.

## Reagan Didn’t Want Communists Getting Computers

This episode’s “Random Access” segment, recorded in March 1984, was presented by Susan Bimba of KCSM-TV for the second time.

- The U.S. Department of Commerce moved to stop exports of computer technology to Soviet-controlled eastern Europe. Specifically, the Department placed new restrictions on Digital Equipment Corporation. The company would now need to obtain individual export licenses for each “sophisticated computer” that it shipped to West Germany, Austria, or Norway–three countries that were known to divert western technology to the Soviet bloc.

- Indeed, despite lobbying from the technology industry, Bimba said President Ronald Reagan had given the U.S. Department of Defense the authority to "advise" the Commerce Department on high-tech exports to all foreign countries. The Pentagon already had that advisory authority with respect to communist countries.
- Following up on an earlier "Random Access" story, Bimba said National Semiconductor was faced with a ban on making chips for the U.S. government after admitting to "inadequate testing" of its products. Despite that, the company reported a $15.4 million profit for the last quarter.
- IBM said it planned to ship more than 2 million personal computers in 1984, nearly triple its output for 1983.
- IBM's newest machine–the portable IBM 5155–led clone-maker Compaq to announce it would cut the price of its Compaq Portable from $2,995 to $2,495. This would allow Compaq to undercut the IBM portable's expected retail price of $2,795.
- Atari said it would lay off 300 more workers, although it planned to expand its sales and engineering staff. Bimba noted that in 1983, Atari lost $500 million, laid off 2,000 workers in the San Francisco area, and moved part of its operations to Taiwan and Hong Kong.
- Another follow-up: After the Japanese Diet considered legislation to reduce copyright protections for U.S. software companies in Japan, the United States Government warned it would retaliate if the measures passed.
- Finally, Paul Schindler reviewed *Ken Uston's Professional Blackjack*. Schindler said the $70 program would "teach you how to play the game and win using various point-counting methods." He also noted the "crisp" graphics and the use of sound.

## John McCarthy (1927 - 2011)

John McCarthy died in 2011 at the age of 84. He was not simply a pioneer in the field of artificial intelligence. He actually coined the term itself during a 1956 workshop at Dartmouth University. Nils Nilsson wrote a short memoir about McCarthy following the latter's death. Nilsson recounted McCarthy's explanation of how he came up with the term:

> "I had to call it something, so I called it 'Artificial Intelligence,' and I had a vague feeling that I'd heard the phrase before, but in all these years I have never been able to track it down."

McCarthy spent time at Princeton, Stanford, MIT, and Dartmouth, before moving to Stanford permanently in 1962, where he remained on faculty until he retired in 2000.

As noted in the episode, McCarthy's most tangible contribution to AI and computer science was his invention of the Lisp programming language, which is still in use today. Nilsson's memoir briefly described how Lisp came about:

> McCarthy spent the summer of 1958 at IBM, and it was there that he first began to recognize the need for a new programming language–one that could support recursion and dynamic storage. (In recursive languages, a program can invoke a [usually] simpler version of itself.) When he returned to MIT in the fall of 1958, he began work on a new language, one he called "LISP" (for "list processor"). Besides recursion, programs written in LISP could process arbitrary symbolic structures and could treat programs (written in LISP) as data (expressed lists in LISP).

Nilsson also described McCarthy as "scrupulously honest in his claims about his work." Noting that AI is often accused of "overselling itself"–Stewart Cheifet would probably agree–Nilsson said McCarthy's papers and memos often contained "modest qualifications" regarding the technology.

## Nils Nilsson (1933 - 2019)

Sadly, Nils Nilsson passed away a few years after McCarthy, in 2019 at the age of 86. According to an obituary by Sophia Kunthara in the *San Francisco Chronicle*, Nilsson worked at SRI International–where Herbert Lechner was a co-worker–for 23 years, serving as its leader from 1980 to

1984. In 1985, he became chairman of Stanford's computer science department, a post he held until 1990. He then served as the Kumagi Professor of Engineering until his retirement and, according to Kunthara, he "helped transition the department from the School of Humanities and Sciences to the School of Engineering."

## Perrone Followed the AI Trail to Apple

The other two in-studio guests from this episode are still alive as of this writing. Jeffrey Perrone continued to work as a self-employed consultant on expert systems well into the mid-1990s. After spending five years as a senior product manager for E*Trade, Perrone co-founded a now-defunct website called PearlSoup.com, and served as its product manager from 2002 to 2009. During the 2010s, Perrone seems to have bounced between consulting and startup work. In 2020, he joined Apple as a contract "conversation user interface designer." (I assume that means he's working on Siri.)

The subject of Perrone's product demonstration, *Expert-Ease*, was written by Donald Michie, a British AI researcher who passed away in 2007. Scott Mace provided a detailed write-up of *Expert-Ease* for the March 19, 1984 issue of *InfoWorld*. The software cost $2,000 and required an IBM PC with at least 128 KB of RAM. (For reference, a stock PC model only came with 16 KB.) Mace said that as of his report, "about 60 sites worldwide have bought early versions of *Expert-Ease*." He quoted Perrone as promising a "new, faster, and easier-to-use version of the program will be released this spring [1984] as it goes into wider circulation." Apparently, the first version of *Expert-Ease* was considered too slow, owing to the fact it was written in Pascal.

## Kehler Continues to Promote AI Decision-Making

As for Thomas Kehler, he has served as CEO of a number of tech companies, including IntelliGenetics. According to a corporate history by FundingUniverse, Edward Feigenbaum started IntelliGenetics to monetize the AI research he was doing at Stanford. The company introduced its KEE system in 1983 as a “general purpose AI tool that had the potential to develop knowledge-based systems with commercial and industrial applications.” Later that year, IntelliGenetics went public, changing its name to IntelliCorp in 1985.

But as FundingUniverse explained, IntelliCorp struggled to capitalize on the potential of artificial intelligence. Ironically, it was the use of Lisp that made the KEE system less attractive to potential customers:

> IntelliCorp, like many of the early AI companies, relied on the LISP programming language, which in turn required a specialized computer that cost as much as $150,000 and could only be used by a single operator. The research and development units of major corporations were willing at first to invest in this technology, thus spurring initial growth in the industry, but the broader corporate marketplace failed to follow suit. It wanted AI programs that could run on its mainframes and minicomputers and did not require its people to learn LISP. As a result, AI advances began to be incorporated into general computer programs, and AI lost its status as a distinct market.

Kehler replaced Feigenbaum as chairman and CEO in 1988 and remained with the company until 1991. Since then, he’s been involved with more than half a dozen tech startups. Kehler is currently CEO of CrowdSmart. He explained his latest venture’s purpose in a 2020 Medium article:

> Whenever capital is deployed, one or more individuals make a judgement call to invest or pass. In most investment scenarios there is an investment committee that makes the call to deploy capital. Our project focused on seed and series A investments where there is little to no reliable predictive data and the process is highly reliant on human judgement.
>
> The project was initiated as part of a new company, CrowdSmart, founded in 2015 to create a technology platform that would change the way capital is deployed to startups. (My co-founders were Kim Polese, Markus Guehrs and Fred Campbell). We believed that a scalable, trusted investment rating (like a Moody’s rating or a FICO score), and

guidance would level the playing field to create more equitable opportunities for startups and better outcomes for investors.

As for IntelliCorp, it sputtered along for a couple more decades after Kehler's departure until its software assets were finally sold in 2019 to Austria-based software company Tricentis.

## Notes from the Random Access File

- This episode is available at the Internet Archive and has a recording date of March 16, 1984.
- Edward Feigenbaum will appear in a later *Computer Chronicles* episode, and I'll discuss him in more detail at that time.
- Stewart Cheifet's demonstration of a small chess computer may have been another nod to the legacy of John McCarthy, who according to Nils Nilsson helped to develop one of the first chess-playing computer programs in the early 1960s.
- The use of a computer AI to make decisions regarding a nuclear reactor reminded me of a similar scenario involving Homer J. Simpson some years later.
- Speaking of *The Simpsons*, I wonder if the people who made *Ken Uston's Professional Blackjack* were also behind *Lee Carvallo's Putting Challenge*.
- *Ken Uston's Professional Blackjack* was not just a simple blackjack game. As Paul Schindler mentioned, it taught you how to count cards, which will get you kicked out of a real casino. (Schindler noted Uston himself had recently been ejected from 5 casinos.) What fascinated me was that the game's settings also adjusted for different types of casino rules. I never knew that blackjack was played differently on the Las Vegas Strip differed than in downtown Las Vegas casinos.
- So who the heck was Ken Uston? Schindler described him as a former stockbroker turned professional gambler. An obituary in *Blackjack Forum* said Uston, who was found dead in a Paris apartment in September 1987, had also been "working as a computer consultant for the Kuwaiti government."

# Chapter 21: The Apple Graphics Tablet, SGI IRIS 1400, and Quantel Paintbox

Personal computers of the early 1980s were often limited to just a few colors for on-screen graphics. The Apple IIe, for example, could display up to 16 colors at one time depending on the screen resolution. And of course, no home computer of this era could produce genuine 3D graphics. That capability was limited to very high-end machines designed for industrial or commercial use.

## The Special Talents of Computer Graphics

Which brings us to our next *Computer Chronicles* episode from 1984. The subject is computer graphics. And while Stewart Cheifet does open the program with a demonstration of a graphics peripheral designed for personal computers, most of the program is devoted to technology that was beyond the financial or technical capacity of the home user.

Cheifet, joined by returning guest host Herbert Lechner, showed off the KoalaPad, a graphics tablet designed for most popular microcomputers of the day. In this case, it was hooked up to an Apple IIe. Cheifet demonstrated how you could simply draw on the screen using the tablet. He said that computer graphics were often associated with drawing or video games, but they were also used in more serious applications. Lechner concurred, noting that computer “fine art” was now hanging in galleries right alongside traditional art. And in the commercial area, Lechner said computer graphics systems aided animators and played an important role in producing television advertising.

The B-roll for this episode included footage of Ron MacNeill, a computer artist with MIT’s Visible Language Workshop. Cheifet explained how MacNeill used a digitizing tablet to choose from a palette of colors and pre-programmed images to create a collage that existed only in the memory of a computer. MacNeill started with materials from different sources, such as photographs and 3D objects, to make a composite that could then be manipulated to change their colors, sizes, and geometric aspects. The final hardcopy was then printed using a plotter to create a wall-size mural that was 14 feet high and 48 feet long.

Cheifet said the “special talents” of computer graphics were also increasingly used in another area of the visual arts: animation. Once again, an artist could use a digitizing tablet to create a figure by drawing in just enough points to determine its size and rough shape. For a symmetrical object, Cheifet said all the computer needed was an outline of one side, which was then shaped into a fluid form, regenerated, and filled with color and other 3D details. The software could then calculate and reproduce the characteristics of light and shadow that the object would possess in three dimensions. Finally, to animate the drawing, the artist could specify key camera positions from which the object was viewed in space. The program then filled in the missing frames to simulate movement between those points. Cheifet said that making a drawing come to life was “the most tedious part” of an animator’s job–but fortunately, it was also readily adaptable to digital processing.

Over some computer graphics demos produced by Pacific Data Images (see one below), Cheifet emphasized that the systems used to produce these images were not designed to replace artists but to assist them. And with enough human talent, he said, artists could use computers to mimic reality in greater detail or expand it to the fringes of their imagination.

An early computer graphics demo by Pacific Data Images, showing a cup filled with crayons and a corked blue flask.

# It Takes a Lot of Computing Power to Move a Rubik’s Cube

Michael Arent and Donald McKinney joined Cheifet and Lechner for the first roundtable segment. Lechner asked Arent, a digital artist and consultant with Aaron Marcus + Associates, for a demonstration of the work he did using an Apple Graphics Tablet. Arent showed you could use the tablet to not only paint images, but also combine graphics, photographs, and text onto the screen. Specifically, he took a digitized photo of Cheifet, removed the background, and added a word balloon with text.

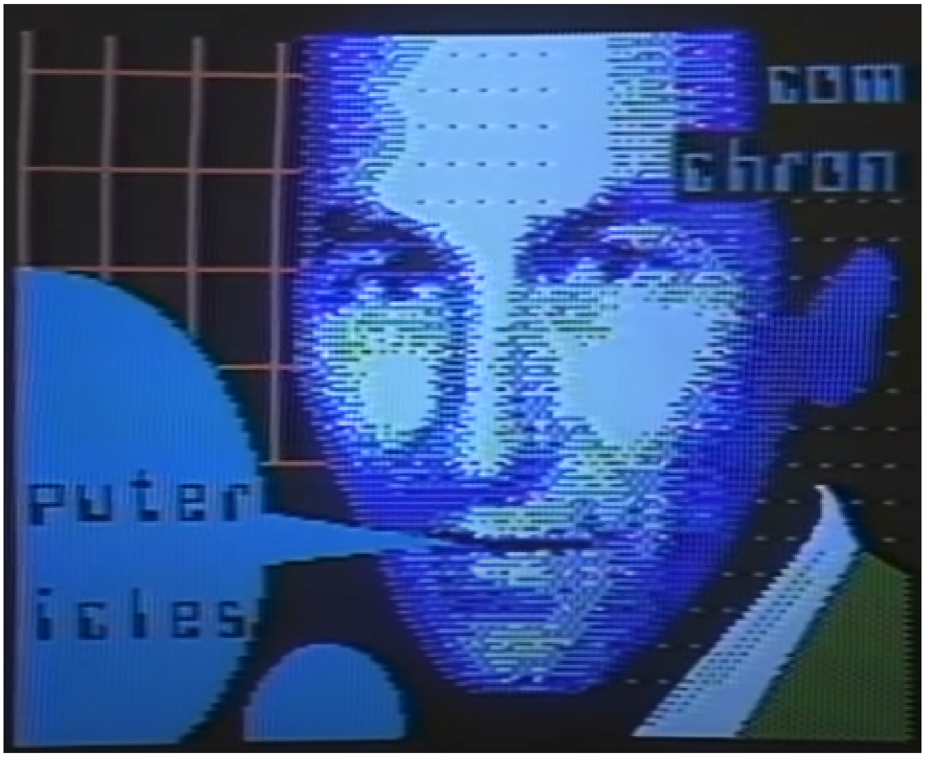


Image of Stewart Cheifet with a word balloon saying “Computer Chronicles,” made by Michael Arent with the Apple Graphics Tablet.

Cheifet pointed out that computer graphics demanded a great deal of memory. How was that accomplished on a microcomputer like the Apple IIe? McKinney, a vice president with Silicon Graphics, Inc. (SGI), replied that it depended on the amount of color that you had available as well as the screen resolution. The higher and smoother that the image appeared, the more memory memory you required, McKinney noted that SGI’s own system, which he would demonstrate in a moment, required approximately 3 million bytes of memory to store an image in real time.

Returning to Arent's demo, Lechner asked how it was possible to remove the background of an image. Did it involve tracing around the image? Arent said yes, the Apple Graphics Tablet had a function that allowed him to "block out" the background by painting it black. Cheifet asked for more specifics on how the tablet worked. Arent said it directly addressed various areas within the Apple IIe's memory, essentially commanding it to put color or geometric figures onto the screen.

Cheifet and McKinney then moved over to the other side of the studio to demonstrate SGI's IRIS 1400 computer graphics system. Cheifet asked how this differed from the Apple Graphics Tablet or the KoalaPad. McKinney explained that those devices worked with microcomputers. But the IRIS was a different class of machine. The main difference was that it used custom chips developed by SGI to perform 3D calculations. He then provided the first demo, which was a Rubik's cube moving in 3D space.

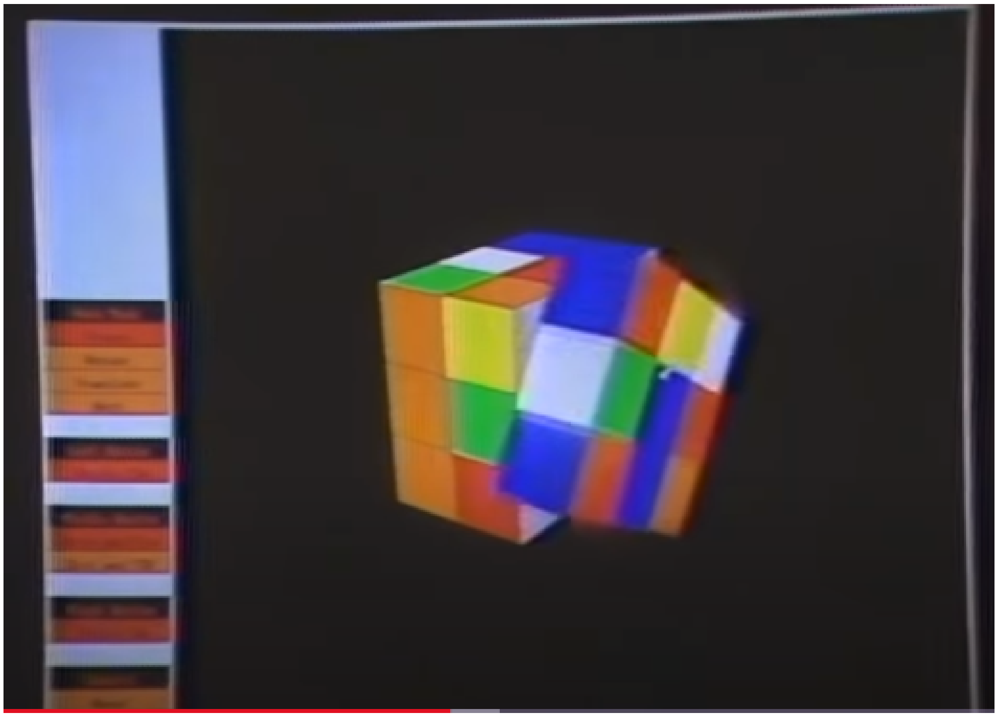


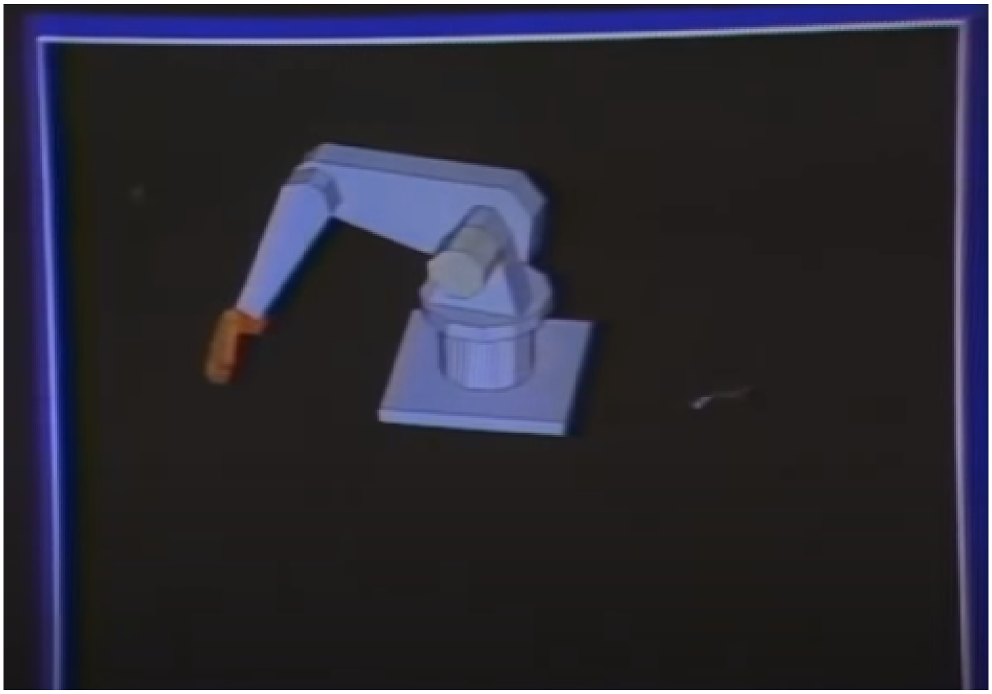
Image of a Rubik's cube moving in 3D space, created by the Silicon Graphics IRIS 1400 system.

Cheifet asked about the hardware used in the IRIS 1400. McKinney said the CPU was a Motorola 68000, which was also used in many high-performance microcomputers (such as the Apple Macintosh). He said the other key was the Geometry Engine, a propriety SGI high-speed 3D floating-point calculation unit that could perform 6.5 million operations per second. In the near future, McKinney said SGI planned to increase that speed to 10 million floating-point operations per second.

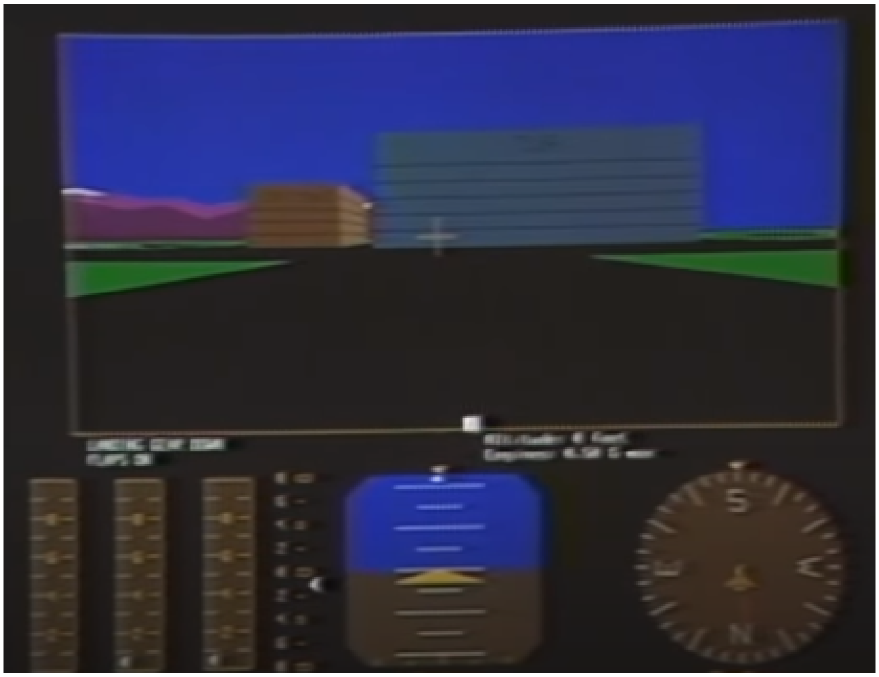
McKinney's next demo was a virtual fly-by of some 3D-rendered buildings. Cheifet asked about the practical application of this technology. McKinney said SGI had several customers in the architectural, engineering, and construction fields. Indeed, many local companies in the Silicon Valley region used the IRIS system to create piping diagrams and perform calculations for intersections in facilities like nuclear power plants. It was also possible for an architectural firm to add their own software to the IRIS system and sell it to end-user architects for use in designing exteriors and landscapes, or even interior decorating.

3D rendering of buildings in a cityscape, created by the Silicon Graphics IRIS 1400 system.

There were two additional demos, one showing a 3D robotic arm and the other a flight simulator.



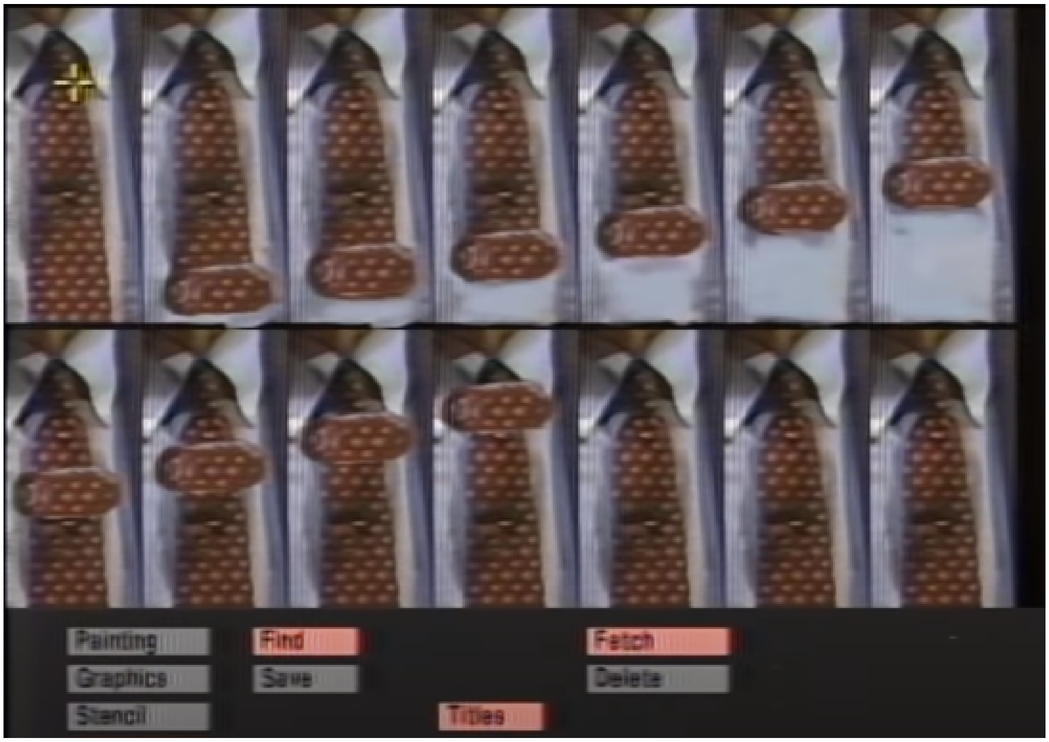
3D rendering of a moving robotic arm, created by the Silicon Graphics IRIS 1400 system.



3D rendering of a flight simulator, created by the Silicon Graphics IRIS 1400 system.

# The Wave of the Future for Television Graphics

For the second and final roundtable, Ann Chase and Kevin Prince joined Cheifet and Lechner. Chase, a freelance digital artist, demonstrated the Quantel Paintbox. She used the Paintbox to create a digital animation of Stewart Cheifet's necktie rolling up and down during the transition to this segment.



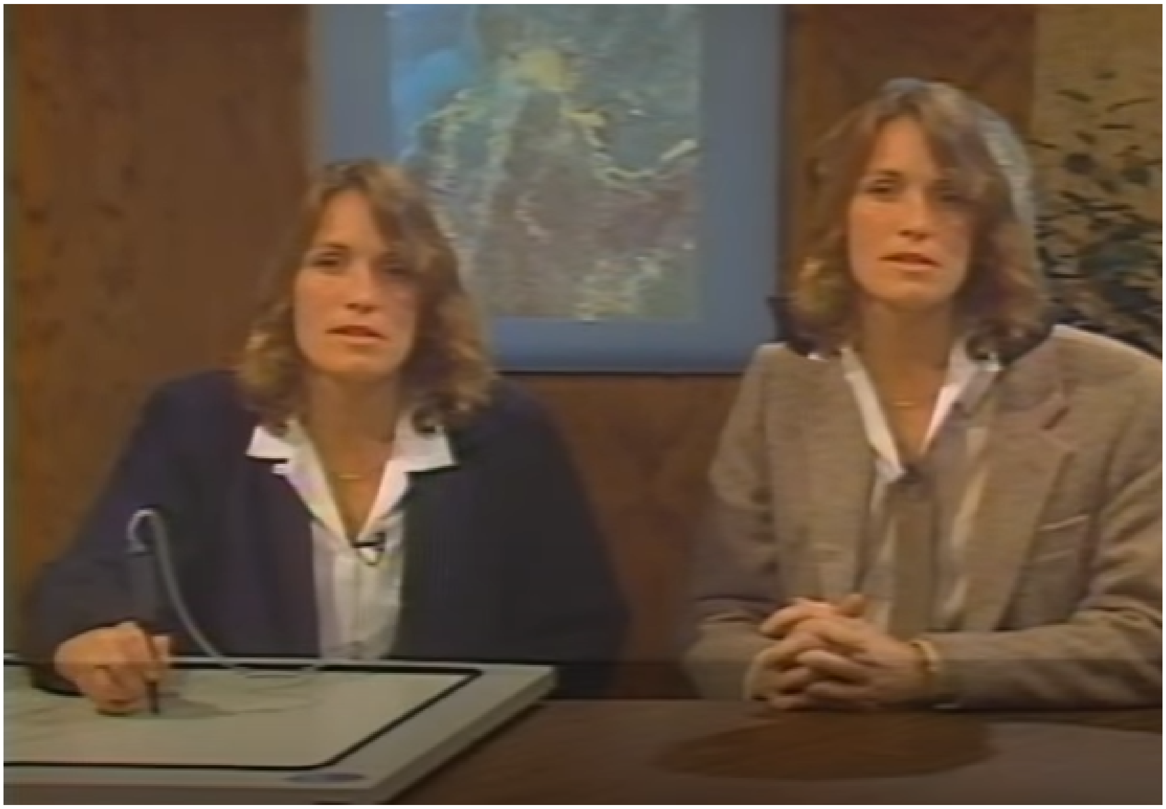
Key frames for the animation of Stewart Cheifet's tie, created by digital artist Ann Chase on the Quantel Paintbox.

Cheifet asked Prince, an engineering manager with Quantel, about the hardware and software capabilities of the Paintbox. Prince said like the SGI IRIS system, the Paintbox also used a Motorola 68000 CPU. Quantel also included a lot of dedicated hardware to provide capabilities like fast wipes, painting, changing brush sizes, and so forth. He added this system required a very large operating system that took Quantel quite a bit of time to develop.

Chase then briefly showed how you could draw freehand using the Paintbox. Chiefet observed that this was much finer drawing at a greater resolution than what the products demonstrated earlier could achieve. Prince explained the Paintbox relied on various methods of mapping and storing images that differed from "individual pixel" operations that those other machines used.

Cheifet noted that the Paintbox was currently used to produce television graphics. Prince said most of the television networks actually had “several” systems, which were primarily used for daily news production. He added the graphics technology was so good that most viewers didn’t even notice.

Chase then performed a final demonstration, showing how you could “cut and paste” images with the Paintbox. Taking a still image from the live video feed of the *Computer Chronicles* taping, Chase pasted a duplicate image of her own head on top of Prince’s body.



Quantel Paintbox demo by digital artist Ann Chase, showing the artist and duplicate of her head superimposed on the body of Quantel manager Kevin Prince.

Lechner said that while it was clear this type of computer graphics technology was valuable to the television industry, what about fine art? Was “computer art” growing and catching on? Chase said definitely–computer art was the “wave of the future.” Lechner jokingly asked if paintbrush manufacturers should be worried.

Finally, Cheifet asked how far could we go with this technology? Would we get to the point where we–meaning people like himself and Lechner–could be replaced by animated people who were synced up with spoken words? Prince said he sincerely hoped not. At the moment, Quantel provided the means for an artist to extend their capabilities. The idea was to turn a reasonable artist into a good one, and an extremely good artist into a superb one. All the technology did was allow the artist to complete their work in a much shorter space of time.

## McKinney Makes IT Big in Network Services

Of today's four guests, the one who has likely seen the most financial success is Donald McKinney. McKinney remained with Silicon Graphics as its vice president of sales until 1987. In 1992, he founded International Network Services (INS). Ostensibly a consulting company, INS became known as *the* temporary staffing agency for network engineers in Silicon Valley during the 1990s, according to a 1998 profile in the *New York Times*:

> I.N.S., founded in 1991, does not manufacture equipment. It does not run computers. It does not re-engineer business processes. It does not develop corporate strategies. What I.N.S. does do is design, manage and secure communications networks. And the company does it so well that it has become the network engineers' engineer – the expert that the network experts can turn to when they need a specialist's touch.
>
> Besides Bell Atlantic, the other telecommunications and technology giants that have used I.N.S. include AT&T; MCI Communications and the partner in its recent merger, Worldcom; SBC Communications; Sprint; BellSouth; Cable and Wireless; Airtouch; GTE; Cisco Systems; Compaq Computer and Oracle.

McKinney had stepped away from day-to-day oversight of the company when this article was written. And about a year later, Lucent Technologies acquired INS in a $3.7 billion all-stock transaction, which made McKinney a billionaire, as he still owned about one-third of the company. Since the 1999 sale, McKinney has been a "private technology investor" in companies like Keyssa.

## Prince Makes Chyrons

Kevin Prince left Quantel in 1984, presumably not long after his *Computer Chronicles* appearance. He's continued to work in the television media end of the computer graphics industry. In 2004 he became chief operating officer of Chyron Corporation, whose early computer graphic systems from the 1970s gave rise to the term "chyron" as a generic reference to the "lower third" overlay during news and sports broadcasts.

Chyron merged with a Swedish company, Hego AB, in 2013. Prince remained with the newly renamed ChyronHego as a senior vice president. He left the company in 2015 to become president and CEO of Broadcast Pix Inc., a Boston-based company that "creates integrated production switchers, as well as cloud-based solutions for media management, collaboration, and distribution." According to Bloomberg, Prince served as CEO until August 2019.

## Arent Makes Mark in UI Design

Michael Arent's career took him from producing freelance digital art to designing user interfaces for major tech companies. He joined Apple in 1988 to work on the UI for what became Macintosh System 7. He was also part of the original team behind Apple's QuickTime multimedia framework. After departing Cupertino in 1994, Arent worked as a senior interaction designer for Tele-TV, a short-lived joint venture between several of the "Baby Bells" and the Creative Artists Agency to produce interactive television systems.

Following stints at Sun Microsystems and Peoplesoft, Arent joined SAP Labs in 2005 as vice president of user interface standards. He remained with SAP until 2013, when he moved to GE Software as its director of user interface design. Arent left that position in 2015 and appears to be semi-retired. He's currently an officer and director with the nonprofit Counter Culture Labs, which is described as a "community supported microbiology maker space."

## IRIS Offered Performance for a Hefty Price

We'll no doubt see more from Silicon Graphics in future *Computer Chronicles* episodes, as the company was a powerhouse in the 3D market during the 1990s. For now, this December 2020 retrospective on the company by Cal Jeffrey provided some additional background on the IRIS systems demonstrated in this episode:

> SGI released its first-generation IRIS systems (models 1000 and 1200) in 1984. These were not standalone workstations, but rather raster display units intended to be connected to more general-purpose machines like the DEC VAX. The early IRIS models used a PM1 CPU board, a variant of the one used by Stanford's SUN workstation that David Brown had helped develop. These systems came equipped with 8 MHz Motorola 68000 processors and 768kB of RAM but no disk drive.
>
> Later that same year, SGI would launch the 1400 and 1500. Each machine got a bump up to 10 MHz and 1.5 MB RAM. They also each had disk drives. The 1400 had a 72MB ST-506, and the 1500 sported a gigantic 474 MB SMD-based drive with a Xylogics disk controller.

McKinney never directly identified the machine he used in this episode as the 1400, but given that it had a disk drive–Cheifet kept swapping the disks out for each demo–I'm assuming that was the machine.

Jeffrey noted that while SGI only sold about 3,500 machines total in the IRIS line, each unit cost between $45,000 and $100,000. Indeed, according to an archived 1984 email from a Silicon Graphics product manager, the IRIS 1400 sold for $22,500–but that took into account a 40 percent educational discount. This meant the "retail" price of the 1400 was $37,500.

## Paintbox Dominated the 80s Mass Media Market

Of course, the IRIS 1400 was still a bargain compared to the Quantel Paintbox. A July 1984 article in the *New York Times* described how the three main television networks used the Quantel system to beef up their sports and upcoming presidential election coverage. With respect to NBC, the *Times* noted:

> The network has increased its artistic staff by 30 percent, to more than 50, and invested in no fewer than nine Paintboxes. These devices, which allow an artist to draw directly onto a screen with an electronic stylus, **cost about $150,000 apiece**. (Emphasis added)

The *Times* also cited an ABC executive, who said that while the company once required a "staff of artists" to draw on-screen graphics, "with the Paintbox an artist can come up with a graphic in 15 minutes that used to take two days."

The United Kingdom-based Quantel made its reputation with the Paintbox. Roland Denning, writing for RedShark noted the machine was used in the production of the famous Dire Straits' "Money for Nothing" music video in 1986. But while the Paintbox "was still growing strong in 1990," according to Denning, it was displaced by the middle of the decade by the rise of Adobe's Photoshop software and never recovered:

> Sadly, Quantel was to go the way of all high-end systems based on proprietary hardware and the company began to contract from 2005 onwards. In 2014 it acquired Snell, formerly, Snell & Wilcox, another British high-end hardware-based manufacturer, famous for their Alchemist standards convertor. In 2015 the Quantel name was dropped and what was left of the company became subsumed by SAM (Snell Advanced Media), to be acquired by Belden in 2018 and then incorporated into Grass Valley. The Rio, stripped of its Quantel name, lives on under the Grass Valley brand and claims to be 'the world's only real-time 8K 60p colour and finishing system'.

## Apple Graphics Tablet Faced Early Setbacks

In an unusual reversal, Apple offered the cheapest product demonstrated in this episode. That's really no surprise, given that the Apple Graphics Tablet was a peripheral designed to work with an Apple II. Based on a retail magazine advertisement that I found, the tablet sold for about $120 as of late 1982. Although this was presumably the original version of the tablet, which Apple released in 1979 at a retail price of $650. According to the *Edible Apple* blog, this first model "wasn't exactly a runaway hit as it was subsequently discontinued when the FCC found that it caused radio frequency interference problems."

The model seen on *Computer Chronicles* came out in 1983 and resolved the FCC's technical concerns. Nevertheless, it may have come to market too late. As another Apple blogger, AppleToTheCore, wrote in a 2013 post:

> The Apple IIe was in full swing and the Macintosh with its full graphical user interface was picking up steam. Not to mention both machines had a mouse. Who needed a tablet?

# Notes from the Random Access File

- This episode is available at the Internet Archive, which provided a date of April 5, 1984.
- The YouTube channel RMC - The Cave managed to find a collector who owns some of the few Quantel Paintboxes still in existence.
- Pacific Data Images, whose early computer graphics work was featured in the B-roll, went on to do the first 3D animation for “The Simpsons” in 1995 and later became DreamWorks Animation.
- Aaron Marcus and Associates, the consulting firm that employed Michael Arent at the time of this episode, is still in business. In fact, Arent was the first employee hired by Aaron Marcus when he started his firm in 1982.
- Unfortunately, I was not able to find any information about the fourth guest in this episode, Ann Chase.
- The IRIS 1400 was a whale of a machine. The main cabinet weighed 200 pounds. The monitor and peripherals added another 100 or so.

# Chapter 22: Dialog and The Source

The ostensible topic of this next *Computer Chronicles* episode was databases. But what we're really talking about here are early online information systems–that is, the precursor to the modern Internet. This was a time (1984) when just getting online was a chore. First, you needed a personal computer with a separate modem peripheral connected to a telephone line. Then you needed a subscription to an online service. The service itself charged you by the minute or hour for access–and that was on top of any long-distance phone charges you might incur if the service did not have a local number.

## More Than an Electronic Filing Cabinet

As the episode begins, Stewart Cheifet and Gary Kildall demonstrated the online shopping feature of the Source, one of the two main subjects of the show. Using an IBM Personal Computer, Cheifet navigated text-based menus to explain how someone could purchase a VCR online using the Source. He noted the online shopping database displayed the machines available in the store's inventory and could provide a description of any of the models. If Cheifet decided to buy a VCR, the service would automatically charge his credit card.

Cheifet explained that it was hard to define the subject of online databases, which were also referred to as "telecomputing" or "information utilities." Cheifet said another phrase commonly used to describe these services was "timesharing." Kildall pointed out that was actually an older term that referred to many people sharing the use of a larger computer system. He said that personal computers had stolen away a lot of customers from timesharing as the newer microcomputers were less costly and more convenient. But he added that personal computers did not compete well with larger systems when it came to accessing large information databases. In that sense, timesharing could make a comeback in the form of information services, which allowed personal computer or terminal users to access a larger database, provided it was cheap enough to use.

Cheifet transitioned into the B-roll, noting there were about 2,000 online databases in active use, most of which served specialized interests. He said the key to a useful and usable system was information selection–i.e., the ability to search through massive amounts of data, locate multiple sources and references, and relate them in a number of different ways. An effective database search could squeeze an hour (or even a month) of research into just a few minutes.

The B-roll itself focused on the use of a specialized database for crop genetics. Cheifet explained that in biotechnology, plant genes were manipulated to produce new varieties in which desirable traits from several plants were combined. Because of the extreme complexity–and often experimental nature–of this research, Cheifet said researchers relied upon a worldwide bank of information. Specifically, a chemical database enabled researchers to look for any documents referring to a particular chemical as it affected the reproductive abilities of wheat. The chemical's reference number was listed in a separate database that included records for 5 million chemicals. When cross-referenced with the document database, the researcher could find studies dating back in 5-year increments and displayed a summary of each study.

But Cheifet added that to characterize a database as merely an "electronic filing cabinet" was to underestimate its power. Indexing and cross-referencing the ideas of a thousand different minds meant a database could offer its users an organized way to connect and relate ideas. Used to its full potential, Cheifet concluded, the database could help a researcher find solid support for a vague theory and bring even distantly related ideas together.

## Electronic Mail: The Important Adjunct to Online Databases

Back in the studio, Jay Fitzgerald and Paul Schindler joined Cheifet and Kildall. Fitzgerald was the west coast regional manager for the Source. Kildall asked what a user could access with this service. Fitzgerald quipped it would give you access to the Source Telecomputing Corporation's computers. He elaborated that the Source was a general, broad-based information retrieval service that mostly focused on current events.

Fitzgerald demonstrated the Source's menu-based system. He showed off the service's electronic mail capabilities. He added the service also provided a bulletin-board system, conferencing capabilities, and even the ability to send messages to non-users through either Western Union's Mailgram or the U.S. Postal Service's E-COM systems.

The actual demonstration of electronic mail proved rather slow, as it required connecting remotely to the Source. Kildall joked this was reminiscent of the old timesharing days. In the middle of the demo, Fitzgerald kept getting pinged by another user who wanted to initiate a live chat. Kildall also pointed out that electronic mail was not really a "database" service. Did the Source actually provide database access? Fitzgerald conceded that the Source did not run its own databases. Rather, it offered users access to the United Press International newswire content service, abstracts of business journals, and Media General's database of historical corporate stock data.

Paul Schindler chimed in at this point, noting that electronic mail was in fact an important adjunct to online databases, because useful information wasn't limited to data generated by someone you didn't know. Citing his own journalism work, Schindler noted his publication, *Information Systems News*, had a Tokyo-based correspondent. And sometimes her information was more useful to Schindler than what he could get from the UPI newswire via the Source. So in that sense, electronic mail was just as essential for him accessing information, in this case from the correspondent. Kildall added that electronic mail was also one thing that could not be done with an isolated personal computer; it required an online service.

Cheifet asked Fitzgerald about the most popular services on the Source. Fitzgerald replied the heaviest use was for communications services, such as electronic mail and the post (bulletin board) system. After that, the next-heaviest usage was for the UPI newswire, and travel-based services, such as looking up airline schedules and making reservations. Schindler asked for clarification about the bulletin board system. Fitzgerald said there were about 100 categories. For example, an IBM Personal Computer user could post a message to a category focused on that machine and connect with other users who could respond to questions about the system, look at hardware for sale, and so forth.

Kildall asked about travel services–could he actually get a written ticket sent to him by making a reservation online? Fitzgerald said not directly through the airlines. Rather, the Source had its own online travel agency that users could access to order airline tickets and make hotel or rental car reservations. Depending on the time constraints, the user could then receive their tickets in the mail or pick them up at the airport.

Cheifet noted that there were a lot of complaints from users of services like the Source–and its rival, CompuServe–about long response times and complicated interface commands. He asked Schindler about his own experiences in this regard. Schindler agreed that the most “fertile area” for making online database services more accessible was to make the interfaces easier to use. In particular, he said there was a burden in remembering long user ID numbers and passwords. He noted that the personal computer had made this easier, as he could store that information on the machine and call it up with the push of one button. That gave personal computers the advantage over dumb terminals when it came to online access. That said, he noted that many online databases, such as Dow Jones, still required memorizing cryptic text commands to retrieve information.

As for response time, Schindler said that was a problem that could be solved by throwing money at it. He noted that the Source and another rival, Dialog, had worked hard to increase the amount of equipment on their end so that response times were fairly reasonable. But there was still a “peak load” everyday when demand could not be met quickly enough. Fitzgerald pointed out that the heaviest usage for the Source came between 6 p.m. and 9 p.m. eastern time, when virtually the entire country was online during their night-time leisure hours.

## A 140 GB Peripheral for Personal Computers

In the second and final roundtable, Roger Summit joined Cheifet, Kildall, and Fitzgerald. Summit was president of the aforementioned Dialog Information Systems. Summit explained that Dialog was an interactive computer-based information retrieval service. Their objective was to store on the computer basically all information that had been published within the last 10 years or so. Dialog started out as an institutional service for librarians and information specialists at major companies. But the personal computer had started to open the service up to individual end users.

Summit then demonstrated Dialog’s ability to search for article abstracts. He searched the National Library of Medicine’s MEDLINE database for articles that reference both laetrile and cyanide. Summit noted MEDLINE indexed 3,000 journals and contained a total of 4 million abstracts. In the case of his laetrile-cyanide search, MEDLINE returned 16 items. The user could then look at each abstract, and if they wanted to view the full article they could push a button to have a printed copy mailed to them.

Kildall asked about the other databases available on Dialog. Summit said there were databases for virtually every area of business, medicine, engineering, science, and technology. Kildall asked how up to date these databases were. Summit said the professional societies responsible for the databases collected literature and send them to Dialog on tape. Dialog then updated the online databases on a daily, weekly, or in some cases monthly basis.

Kildall asked about the “freeform” nature of the language used to query the database. Summit said it basically relied on “AND OR NOT” logic gates combined with keywords. Kildall also asked about the cost and convenience factor with a system like Dialog. Summit replied that in terms of using Dialog, it essentially functioned as a peripheral of 140 GB to the personal computer at the end of a telephone line. Kildall mentioned the actual cost of using the service. Summit said the average query would run the user anywhere between $2 and $5. He added that Dialog also offered an after-hours service that included the company’s principal databases for 40 cents a minute, which was cheaper than the regular daytime service.

Cheifet asked about the future of online databases–would we get to the point where such systems might include graphics? Summit said the immediate next step was offering complete source documents, i.e., the full text of articles as opposed to just abstracts. He said there was already a major project underway to accomplish this. Right now, Summit said Dialog currently offered the full text of the Harvard Business Review online, and the UPI newswire was coming soon.

Kildall then brought up the possibility of video discs–which offered up to 1 GB of storage–emerging as a competitor for database services on personal computers. Summit conceded there might be a market for video discs in specialized areas where there was intense interest. But with respect to more generalized databases like Dialog, the coverage was so comprehensive, and the use density so low, that even compared to video discs it was still more cost-effective to store that information centrally and allow users to access it when needed.

Cheifet then reiterated his earlier point about user complaints over slow response times. Would online services really be able to keep up with demand? Fitzgerald admitted there was certainly a technical limitation. But the Source had ramped itself up to such a speed where it could now anticipate and keep ahead of demand.

Cheifet then raised another possible competitor to telephone-based services in the form of cable. Fitzgerald said at this point, he did not see much competition coming from cable, although it would likely play a more important role as time went on and more cable systems were installed throughout the country. But for now, telephone-based systems were satisfying the needs of customers.

Kildall pointed out, however, that telephones only worked well in areas where there was a local interface. He lived in the Monterey area, where accessing online services usually required paying high long distance charges. Fitzgerald replied that the primary networks used by the Source–and he believed by Dialog as well–were expanding on a weekly basis, with new nodes being installed in more remote locations.

Cheifet concluded by asking if it would be possible in the future to see a merger of database management software and a service like Dialog, perhaps in the form of artificial intelligence that could identify a needed piece of information locally and retrieve it remotely. Summit said he could see things moving in that direction, particularly in specialized areas like pulling together a dossier on a person, product, or company.

## The Source Promised to Launch the Information Age, But Quickly Fell Back to Earth

Of the two online services discussed in this episode, the Source was clearly the less successful. Indeed, the service's parent company, Source Telecomputing Corporation, had a fairly tumultuous run. Jimmy Maher published an extensive history of Source Telecomputing for his blog, *The Digital Antiquarian*, in 2017, which began with the story of the service's high-profile launch:

> On July 9, 1979, journalists filtered into one of the lavish reception halls in Manhattan's Plaza Hotel to witness the flashy roll-out of The Source, an online service for home-computer owners that claimed to be the first of its kind. The master of ceremonies was none other than the famous science-fiction and science-fact writer Isaac Asimov. With his nutty-professor persona in full flower, his trademark mutton-chop sideburns bristling in the strobe of the flashbulbs, Asimov said that "this is the beginning of the Information Age! By the 21st century, The Source will be as vital as electricity, the telephone, and running water."

As Maher noted, however, Asimov wasn't the only colorful character present. There was also the founder of the Source, William Von Meister, whom Maher described as "high-strung, mercurial, careless, [and] sometimes a little unhinged." Von Meister, who died in 1995, would be remembered primarily for another startup he founded in 1983, Control Video Corporation, which was the predecessor to America Online.

The Source, meanwhile, sputtered out of the launch bay. Maher said that by March 1980, the service "had managed to attract about 3,000 subscribers, but the online citizens were growing restless," due to a lack of promised features and reliability issues. The Source's electronic mail system, for example, was "down about half the time," according to Maher, "and even when it was allegedly working it was anyone's guess whether a message that was sent would actually be delivered."

Not surprisingly, the parent company, originally called Digital Broadcasting Corporation, was also bleeding money. Maher said Von Meister's business partner, Jack Taub, had "little awareness" of "what Von Meister was doing with the company's money." Eventually, Digital Broadcasting defaulted on a government-secured loan, which forced the United States Department of Commerce to pay $3.2 million to a North Carolina bank. Under threat of government seizure, Taub agreed to give the government $300,000 upfront and repay the rest of the $3.2 million over the next 22 years.

Taub then quickly moved to sell the now-renamed Source Telecomputing Corporation to The Reader's Digest Association–yes, the people who publish the magazine. This was then complicated when Von Meister sued Taub after the former was forced out of the company. Eventually, the Reader's Digest sale went through, and it controlled the Source during the period when it was featured on *Computer Chronicles*.

But new ownership apparently did not turn things around. A July 1982 report in *The New York Times*, said, "Despite heavy investment in the Source, the company so far has not reached the goals the Digest set for it," and Reader's Digest was already "disillusioned with its venture into electronics" and looking to sell. An unidentified source–no pun intended–told the *Times* that the Source was losing between $5 million and $7 million. While Reader's Digest did not immediately bail on the Source, it did sell a substantial stake to Control Data Corporation, a well-known mainframe and supercomputer manufacturer, which came in as an operating partner.

As for the online service itself, the *Times* noted that although the Source was cheaper than most "commercial electronic data banks" of the time, its prices were still considered too high for most home users. Customers had to pay a one-time registration fee of $100 just to obtain the Source software. (A brochure from 1984 showed this fee was later slashed to $29.95.). The user then had to pay an additional monthly fee plus online access rates then varied between $5.75 and $29.75 per hour. In his own report, Maher said the minimum hourly rate had risen to $7.74 per hour by 1984–around the time of the *Chronicles* episode–and that while the service had about 60,000 subscribers at that point, many of them “rarely or never logged on, paying only the minimum monthly charge of $10.”

In March 1989, The Source was sold to, and shut down by, its leading rival, CompuServe. At the time, The Source had about 40,000 subscribers, according to FundingUniverse.

## The Father of Online Systems

In contrast to the Source, Dialog proved to be a much more durable online service. In fact, Dialog still exists. Its current owner is ProQuest, who purchased Dialog–now called Dialog Solutions–from Thomson Reuters in 2008.

Roger Summit founded Dialog while working at Lockheed Corporation’s Palo Alto Research Laboratory. Summit explained in a 2003 interview with Susanne Bjørner and Stephanie C. Ardito for the database industry magazine *Searcher*, that he started experimenting with information retrieval programs at Lockheed in the mid-1960s. These experiments eventually led to the creation of a NASA-funded database system around 1968, which Summit called “Dialog.” Summit’s work later led many to describe him as the “father of online systems,” particularly with respect to search technology.

Lockheed started offering Dialog as a commercial online service in 1972. Summit spun off Dialog as a Lockheed-owned subsidiary in 1981. In 1988, newspaper publisher Knight-Ridder acquired Dialog and renamed the subsidiary Knight-Ridder Information, Inc. It became known as Dialog Corporation after British information services company M.A.I.D., plc, purchased Knight-Ridder Information in 1997. Thomson Reuters then acquired Dialog Corporation three years later.

As for Summit, he remained with Dialog through its many ownership changes as a consultant with the title of chairman emeritus. In the 2003 interview, Summit said he was pursuing the “four major passions in my life: family, technology, music, and sailing,” in addition to serving on the boards of various corporations and non-profit organizations. The most recent public appearance I could find from Summit was a 2005 Google Tech Talk on the origins of Dialog.

## Notes from the Random Access File

- This episode is available at the Internet Archive. There was no definitive recording or airdate, but I would peg this episode around January 1984 based on the email messages seen during Jay Fitzgerald's Source demo.
- Speaking of Fitzgerald, I was unable to locate any information about him or his post-*Chronicles* activities.
- I recently learned that Paul Schindler has a blog!
- On top of all of Source Telecomputing Corporation's other problems, the company faced a trademark infringement lawsuit in 1986 from Source Services Corporation, an Illinois-based company that provided "career guidance and personnel placement services." The case survived summary judgment in a published opinion, but I'm not sure about the final outcome.
- In case you're wondering why Summit used "laetrile and cyanide" for his research example, there was quite a bit of controversy in the 1970s and early 1980s over the use of laetrile–a bitter substance found in fruit pits–as a treatment for cancer. The U.S. government banned laetrile for medical use (over the objections of a number of state governments), because it produces cyanide when introduced into the body.

# Chapter 23: Steve Wozniak, Adam Osborne, Lore Harp, and Gene Amdahl

This next *Computer Chronicles* episode focused squarely on people rather than products. The formal subject is "computer entrepreneurs." And the four guests are people who were all quite well known in the computer industry during the early 1980s. What's fascinating, as we'll see a bit later, is that two of the guests had ventures that each managed to flame out not long after this episode aired.

## "I Had Been Working My Whole Life to Build a Certain Type of Computer for Myself."

Of course, co-host Gary Kildall was himself a well-known computer entrepreneur, having founded Digital Research in 1974. Stewart Cheifet opened the show by asking Kildall about the changes to the "people side" of the computer industry over the decade that followed. Kildall joked he'd traded in his cowboy boots and jeans for a three-piece suit. On a more serious note, he said the biggest change he'd observed was that the industry went from having no products that were competitive to a market where everyone–including IBM–was now producing machines with multi-million dollar budgets. As a result, the industry had become more professional with higher stakes, but also more fun in Kildall's estimation.

Instead of the usual B-roll this week, we actually get clips from a sit-down interview with Apple co-founder Steve Wozniak at his home. (The female interviewer is not identified.) Wozniak noted that back when he started working on microcomputers in the mid-1970s, there were no interface cards, peripherals, or software for such machines. This made things exciting for people like himself as they were doing things for the first time.

Indeed, Wozniak emphasized that around this period–1974 to 1975–he was not even thinking about what steps were necessary to have a successful product or company. Instead, his motives were purely personal: “I had been working my whole life to build a certain type of computer for myself.” And he built the best one–the Apple I followed by the Apple II–that was doable at the time with the components available. Wozniak emphasized the “freedom” that came with this approach, as opposed to working on a company project where some manager defined what needed to be done.

Wozniak said that now–1984–everywhere he went there were meetings with people in three-piece suits. The computer industry was a large business with a lot of dollars involved. The people who were coming out of school trained to run and manage a business were key now. Consequently, Wozniak said there were very few technically creative products being developed in the microcomputer field.

The interviewer then asked Wozniak if a “garage operation” was still feasible in the small computer industry. Wozniak said he thought that era was over with respect to hardware. It used to be that people could quickly manufacture simple personal computer boards on a small budget and sell them by putting an ad in a hobbyist magazine. Today, that was still possible for software groups. If you came up with a good idea for a software project, there were avenues to find companies that could market it for you or turn it into a product.

Wozniak added that about once per decade, a very large market would come up from zero to make billions of dollars within a few years. It was possible this could happen again in microelectronics and computers, he said. He could see a new group of people coming along who weren’t part of an established platform like Apple and start new businesses. These new garage operations could then become very large companies within a few years. Of course, they’d be shunned by the established players, just as Apple was at first. Wozniak concluded with the observation that when you’re looking at someone else, you can understand how you were perceived back in those days and realize they were right–it was just a quirk of fate that they were wrong.

## “There Will Never Be an IBM in Software”

Back in the studio, Adam Osborne and Lore Harp joined Cheifet and Kildall. Osborne was the man behind the eponymous (and bankrupt) Osborne Computer Corporation, which he’d left to start a new venture, Paperback Software International. Kildall reminisced that he and Osborne had worked together with Intel back in 1973. Kildall then asked Osborne if it was still possible to get into the computer industry or if the market had been completely dominated by companies like IBM and Tandy. Was there still a place for a young inventor to make their mark? Osborne replied that it depended on what area of the market you were talking about. If you were going to build a microcomputer, it probably wasn’t a good idea, as there was nothing hi-tech about that market anymore. That space was now a “Clash of the Titans.” Building a computer in this market was like building a washing machine or a refrigerator, Osborne said, in that it was about the economics of volume and reliability.

On the other hand, Osborne said there were more opportunities than ever before in other areas. He analogized it to selling razor blades as opposed to razors. Cheifet asked for clarification–what areas could an entrepreneur make a mark? Osborne said peripherals, software, and other add-ons. He noted you could make a lot of money selling extra cards that fit into a popular computer. As for software, that part of the industry was still in its infancy. Osborne predicted there would be 10-to-100 times more software sold 5 or 6 years from now. Such hyper-growth would create a lot of opportunities.

Cheifet then asked Osborne about who the “Titans” were that he referenced earlier. Osborne said IBM was the main player. He expected AT&T would come into challenge IBM in a big way. He also said at least one of the Japanese computer companies would come in, and together with IBM and AT&T dominate the market.

Cheifet turned to Harp, the CEO of computer manufacturer Vector Graphic, and asked for her assessment. Harp said she largely agreed with Osborne. She said that in her estimation, it would be cost prohibitive for any new computer company to get started today. When she started Vector back in

1976 it was possible to do so with $5,000 in capital (and a lot of sleepless nights). Today that was impossible. A company needed several million dollars in capitalization, and even then that might not be enough money to get critical mass in a short enough period of time to become a contender. But like Osborne, Harp saw room for growth in computer add-ons, such as communications peripherals that linked systems together.

Kildall asked if she meant communications services like the Source or CompuServe. Harp said that was part of it.

Returning to the idea of the "Titans," Kildall noted that companies like IBM only entered an area of software or hardware when it was proven to be successful. This meant the small "garage shop" operations still had opportunities where the larger companies were not willing to put in the effort right now. Osborne added there were also price considerations. When people were spending thousands of dollars on a computer, for example, they tended to be very careful about what they bought because it was an investment. In contrast, if you're buying a $50 product–even as much as a couple of hundred dollars–you're more likely to say "what the heck" and buy it, as even if it doesn't work out you've still learned something. Osborne said that was why you saw a lot of very inexpensive computers now sitting in homes and collecting dust.

For that reason, Osborne declared, "There will never be an IBM in software." Because when you're dealing with $50 software products, the vicariousness of human nature would lead everyone to want to buy something a little different. Kildall agreed, saying that software was where we'd likely see all the real inventors go.

Kildall followed up on Osborne's remark about smaller inexpensive computers gathering dust. Did Osborne have any idea about, say, the percentage of sub-$1,000 personal computers that were still in productive use? Harp actually replied, saying she believed about 50 percent of such machines were now gathering dust. She admitted there was no way to substantiate that figure as there was little research in this area. But she relied on personal anecdotes from people who had purchased early machines from places like Radio Shack. Once these people brought their cheap computers home, they realized they didn't quite have the need they thought they did for the system.

Osborne chimed in, claiming his company had done some research in this area and found it tended to vary by brand. He cited the Sinclair ZX80 and ZX81 machines as examples of computers that were mostly gathering dust. Again, Osborne pointed to the price. People purchased those machines for around $300 just to find out what computers were all about. Having fulfilled that purpose, the fact the machines were now collecting dust should not be considered an indictment of the products. Osborne also pointed to the Commodore 64 as perhaps the model where more machines were collecting dust than any other, primarily due to reliability problems. And given the machine only cost $200 in the first place, customers were just not going to pay to have a Commodore 64 fixed when it broke down.

Harp added that her own company, Vector Graphic, had no interest in getting into the low-cost home computer market. She said anything under $3,000 was an absolute no for her.

Cheifet said there had been chatter about a "falling out" starting to take place in the computer industry. He noted many people considered the demise of Osborne Computer an example. Was this in fact the start of a trend? Osborne said that his former company's failure had nothing to do with competition or an industry "falling out." He said the company "plain and simple committed suicide," although he declined to elaborate at this time. As to Cheifet's question about an industry falling out, Osborne said the answer was partially yes and partially no. IBM, with its 70 percent market share, was clearly dominating the market. On the other hand, if you took that remaining 30 percent, it was huge enough to support a lot of companies that could sell between 3,000 and 10,000 computers per month.

## "I Don't Think Clones Will Be a Long-Term Solution"

Joining the group for the final segment was Gene Amdahl, a legendary former IBM executive who now ran his own startup supercomputer manufacturer, Trilogy Systems. Kildall asked Amdahl for his perspective on the viability of the new IBM Personal Computer clones that were on the market. Could they be successful? Amdahl said he was not that familiar with the PC end of the market, but he was quite familiar with IBM and how it operated.

Amdahl noted the company's pattern had not changed much over the years. They move into a field once it was proven just well enough in terms of becoming a size sufficiently large to be of interest–and where somebody else has learned what it takes to satisfy the market. At that point, IBM moves in to take its share of the action. Assuming IBM has the same degree of success in personal computers as it had in high-performance servers, Amdahl said the IBM architecture would become a *de facto* standard–not because IBM had the best architecture, but rather because so many people would expect them to be successful. Consequently, the cost of developing applications for the PC would be sufficiently great to discourage people from changing to another architecture.

Kildall asked how competitors could track a change or transition that IBM might make in the future on a moment's notice. Amdahl replied you couldn't do it perfectly, but if you were in a portion of the field where the technology or certain functional limits had formed the boundary of where you could go, as those possibilities moved forward, you could expect IBM's offerings would likely move in that direction.

Kildall asked Lore Harp for her feelings about the IBM clones. She said they were not going to have a future. Right now, the clones were enjoying a tremendous wave because of IBM's inability to deliver their machines in sufficient quantities. But most of the clone companies were not creating any value. They were putting hardware together and buying software available to everyone. But they were not adding anything that made them "proprietary." She believed that if IBM took a different turn, the software companies would follow suit and leave the clone makers scrambling to protect their customer base. And she did not see how these competitors could turn as quickly as IBM, since the company planned its moves a year or more in advance. Harp was emphatic, "I don't think clones will be a long-term solution."

Cheifet turned back to Adam Osborne and asked if the computer business was now more driven by advertising and packaging than technology. Osborne said the major difference between the personal computer market today and the high-performance mainframes that Amdahl was talking about was the fact the former really had nothing to do with hi-tech anymore. For that reason he both agreed and disagreed with Harp's assessment of the market. He agreed the bulk of the clone companies would not make it. But there was still a vast market for clone companies.

Osborne broke it down this way: There was about 15 percent of the market that would wait for Hell to freeze over before buying from IBM. Another 15 percent of the market didn't care. And the remaining 70 percent preferred IBM. That 15 percent who would never buy from IBM was still a huge market. Osborne added that IBM was shipping 10,000 personal computers a day right now. Even if IBM ended up leaving some of those customers behind, there was room to come in. So if you got a company who understood manufacturing on a low-cost, high-quality basis, they could start eating into IBM's PC share in a way that nobody ever could on the mainframe side.

Kildall observed this meant the clone companies that would be successful would be the ones that thought of themselves as manufacturing companies rather than computer companies. Osborne said that was absolutely right.

Cheifet brought the discussion and program to a close by asking about the role of venture capital. How was it different now trying to raise funding for a computer company now versus a decade ago? Amdahl noted that during the 1973-1975 recession, when capital gains tax rates were also very high, the stock market for new initial offerings was "abysmal" to say the least and essentially non-functional. Today, there was a rather "buoyant" market. Cheifet asked Harp about her own VC experience. Harp noted she'd been through two rounds of funding, once in 1979 when the capital pool was just starting to grow, and again now in 1984. As far as her company, Vector Graphic, was concerned, they did not have much leverage to determine their price since they badly needed the cash. But in many startup situations, when someone had a good angle, they could essentially name their price because there was not many good deals around, but there was a tremendous amount of venture capital available.

## Adam Osborne (1939 - 2003)

Adam Osborne was a chemical engineer by training. Born to British and Polish parents living in Thailand, Osborne later married an American citizen and moved to the United States. He first became known in the tech world as an author and publisher of computer books, including his 1975 text, *An Introduction to Microcomputers*. From this, Osborne launched his publishing business, Osborne & Associates, which he sold to McGraw-Hill in 1979.

Osborne started his computer company in 1981. Its first and most famous product was the Osborne 1, a 23-pound portable CP/M-based computer with a built-in black-and-white screen. As described in Martin Campbell-Kelly's obituary of Osborne for the *Independent*:

> A unique selling point of the computer was that it came with a software bundle consisting of a word-processor program, a spreadsheet and other popular application programs; in this way users could have the computer up and running without the need to buy additional software. It was said that the retail value of the software was more than the computer's $1,795 selling price.

The Osborne 1 predated the launch of the IBM Personal Computer. And while the machine was initially successful, selling about 100,000 units according to Campbell-Kelly, Osborne quickly faced stiff competition from IBM and other early portable computer manufacturers like Kaypro and Compaq.

During the *Chronicles* episode, Osborne claimed that many people misunderstood the reason for the company's quick demise. Technology writer Robert X. Cringely elaborated on this in a 2007 article. Essentially, the conventional wisdom was that Adam Osborne destroyed his own company by "pre-announcing" a new MS-DOS machine that would replace the Osborne 1's existing CP/M model. This purported marketing blunder–

alluded to even in the *Independent* obituary–led to a collapse in demand for the Osborne 1. But Cringely, who spoke with a former Osborne Computer employee, said the real story wasn't quite so simple:

> [I]n fact, [Adam Osborne] told reporters there wouldn't be a DOS version for at least a year because it was too expensive for his price-sensitive line. He announced the Executive – the follow-on model to his successful first Osborne – one month before delivery, and sales dropped in half for a couple of weeks while inventory cleared out.

Now, demand for both Osborne machines did collapse, but Cringely said the reason for that was the new Osborne Executive "was not competitive with the Kaypro," which he described as a "slapped-together rival" that cost $200 less than the Osborne 1 and came with a built-in screen that was larger than either Osborne model. Basically, all of Osborne's customers went to Kaypro. But this was not the result of Adam Osborne jumping the gun to announce an MS-DOS machine.

In any event, Osborne Computer never shook off the failure of the Executive machine. The company entered bankruptcy in 1983. It emerged from reorganization briefly in 1984 and managed to produce a couple of new models. But by 1985 the company was back in bankruptcy and ultimately liquidated

Meanwhile, Osborne's newest venture at the time of his *Chronicles* appearance, Paperback Software, also ended up suffering an ignominious demise. For all the talk of IBM clones in this episode, Osborne positioned Paperback as a software clone-maker of sorts. Specifically, he marketed a $99 spreadsheet program called *VP-Planner*, which was basically a clone of the popular *Lotus 1-2-3* software, which cost about $500.

Lotus sued Paperback for copyright infringement. The crux of the lawsuit was that Lotus considered the 1-2-3 user interface–notably the structure and organization of its menus–to be copyrightable material. A federal judge in Massachusetts agreed and issued an opinion in June 1990 holding that Paperback committed copyright infringement. The litigation took nearly three years, during which time Paperback's sales cratered and never recovered. The court's decision forced Paperback to withdraw *VP-Planner* from the market and the company dissolved soon after.

Osborne ended up leaving Paperback shortly before the judge's ruling. In 1992, he contracted a brain disorder and decided to return to India, where he'd lived as a child during World War II. Osborne died in India in March 2003.

## Gene Amdahl (1922 - 2015)

This episode didn't really touch on Gene Amdahl's enormous contributions to the computer industry. Jack Schofield, in an obituary of Amdahl for the *Guardian*, offered some crucial context:

> Whenever you draw cash from your bank, take out insurance, book a flight, or shop in a large department store, the transaction was probably handled by an IBM mainframe originally designed by a former South Dakota farm boy, Gene Amdahl, who has died aged 92. Amdahl's System/360 mainframes, announced in 1964, transformed IBM, and transformed computing. They became such an essential part of large-scale data processing that most Fortune 500 companies are still using them 50 years later.

Amdahl left IBM in 1970 and made multiple efforts to start his own competitor to cut into Big Blue's share of the mainframe market. His first venture, Amdahl Corporation, developed IBM-compatible machines. Gene Amdahl then left that company in 1979 to start Trilogy Systems. Amdahl Corporation continued on without him until the mid-1990s, when it was acquired by Fujitsu.

As for Trilogy, things didn't quite go so well. In Part 13 of this series, I discussed a "Random Access" news item about Trilogy delaying the shipment of its promised supercomputers "due to problems with the metal coating of its prototype chips." That delay proved to be permanent.

Myron Magnet, writing for *Newsweek* in September 1986, detailed how Amdahl and Trilogy burned through $230 million in capital–including $60 million raised in a public offering–to develop an integrated chip for mainframes:

> The project kept missing deadlines, then revised deadlines. Harder to fix, the company wasn't working quite right either. The parts of the organization never seemed to mesh precisely. Engineers didn't develop that second-nature reliance on one another that for Amdahl marks a successful technology company. Instead employees galloped off in all directions, painting their own projects as essential and demanding reinforcements. The organization grew bigger and more unwieldy by the month. Like tunnels started from both ends that fail to meet in the middle, the separate projects never cohered into a whole. "We were off only a little," Amdahl says in his slow, shy manner, "but it was enough."

In 1985, Trilogy merged into Elxsi, another computer manufacturer. Amdahl remained with the merged company until 1989. He went on to form two other startups, one of which, Commercial Data Servers, apparently still exists today under the name Xbridge Systems, Inc.

## "Bored Housewives" Started One of the First Small Computer Companies

Vector Graphic started out in the computer industry around the same time as Apple. Indeed, Benj Edwards noted in a 2015 history of Vector for *Fast Company* that Lore Harp and her co-founder, Carole Ely, had a booth at the first West Coast Computer Faire in 1977 just a few rows from where Steve Wozniak and Steve Jobs showed off the Apple II.

Edwards described Harp and Ely as "bored housewives" looking to go into business together. After rejecting the idea of starting a travel agency, they decided to start a computer company after Harp's husband, Bob Harp, had been tinkering with an Altair 8800 computer kit. Bob Harp had developed an 8-kilobyte static memory board for the Altair, which became Lore Harp and Carole Ely's first product. Edwards said Bob Harp suggested the name "Vector Graphic" because he planned to develop a video board that never came to fruition.

As Lore Harp mentioned on *Chronicles*, Vector Graphic only had a few thousand dollars in startup capital–$6,000 according to Edwards. At first the company only sold the memory boards. But later, Bob Harp designed several additional components that were then assembled into a full-scale computer:

> The result was the Vector 1, launched in 1977, which shipped in two case colors, green or orange (or "rust" as they called it at the time). Lore and Carole's emphasis on visual aesthetics led them to offer this choice of colors at a time when many companies gave little thought to what their computers looked like (and it was only starting to become a given that PCs shipped in a case at all). An attempt to order orange circuit boards to match the orange case went awry when the first batch of fifty came back pink.

Lore Harp said that although the Vector 1 and Apple II launched at basically the same time, she saw her company as targeting the business rather than the home computer market. "We always looked at the Apple II as more of a toy," Harp told Edwards. However, when the spreadsheet program *VisiCalc* launched exclusively on the Apple II in 1979, Edwards said that prompted many business customers to prefer Wozniak's machine over the Harps' Vector 1.

Nevertheless, the Vector 1 sold well and Lore Harp was featured in the national media as one of the country's first prominent female computer executives. Edwards said that Harp "would even end up partying in Paris, France, with California governor Jerry Brown and Steve Jobs as part of a delegation of California technologists."

But the business took a toll on the Harps' marriage and they ended up divorcing. Meanwhile, Vector faced an existential threat from IBM's looming entry into the personal computer market. Lore Harp decided to take Vector public before IBM took away their market share. Bob Harp ended up leaving the company after he failed to persuade his estranged spouse to pivot to manufacturing IBM PC clones instead.

Vector did go public and Lore Harp briefly stepped aside as CEO following her second marriage to Patrick McGovern, the founder of computer magazine publisher IDG, in 1982. Harp returned as CEO, however, in May 1983. By this point, Edwards described Harp's role as "trying to chart the destination of a sinking ship." Most of Vector's business customers had switched to IBM. Shortly after her *Chronicles* appearance, Harp resigned as CEO for a second time.

Vector belatedly tried to move towards IBM PC compatibility, but the company filed for bankruptcy in 1985. The following year, a holding company liquidated Vector's remaining assets. Edwards offered a short eulogy: "The truth is that, from 1982 on, no one could have saved Vector. The firm ultimately shared its fate with the every other PC maker that didn't jump on the IBM clone bandwagon."

Today, Lore Harp McGovern chairs the McGovern Institute for Brain Research at MIT, which she co-founded with her late second husband, who passed away in 2014.

## Woz's Apple the Lone "Clone Wars" Survivor

Ironically, the only company represented in this episode that survived the IBM clone wars was Steve Woziak's Apple. Everyone knows that Wozniak designed the Apple I and Apple II computers. He also played an early role in the development of the Macintosh, although that was cut short after Wozniak sustained injuries in a February 1981 plane crash and briefly left Apple. He did return for a stint in the mid-1980s and reportedly remains on the payroll even to this day, but he has not been an active employee since 1985. Wozniak has participated in a number of startup ventures since then, none of which strike me as particularly notable–unless you want to count the Us Festivals.

Wozniak turned 71 on August 11. He was most recently in the news for speaking out in favor of "right to repair" laws, which are strongly opposed by current Apple management.

## Notes from the Random Access File

- This episode is available at the Internet Archive, which has a date of October 5, 1984, although I suspect this episode was recorded several months earlier.
- The unidentified woman who interviewed Steve Wozniak may have been either Janet K. Moore or Holly Murray, who were both credited as segment producers on this episode.
- The recording of this episode also featured yet another series of telecourse interstitial segments presented by Herbert Lechner.
- Five years after a district judge said that Adam Osborne's Paperback Software infringed the copyright of *Lotus 1-2-3* by copying its menus, the U.S. First Circuit Court of Appeals held in a different case involving Borland's *Quattro Pro* spreadsheet that "that the Lotus menu command hierarchy is uncopyrightable subject matter."
- Although everyone's fear of IBM crushing the clones was understandable–given that all three in-studio guests' businesses suffered due to Big Blue's dominance–it obviously turned out not to be a permanent state of affairs. The clones eventually won and IBM failed to pivot the market towards a more proprietary standard (which we'll get to in future episodes).
- Adam Osborne's prediction that there would never be an "IBM in software" proved a swing-and-a-miss in the face of Microsoft's future Windows monopoly, which helped establish the company's IBM-like dominance in a number of software fields, including office suites, where *Excel* eventually crushed *Lotus 1-2-3*.
- Osborne would not elaborate during the episode on the reasons for the failure of his former company, but later in 1984 he did publish an autobiography, *Hypergrowth: the Rise and Fall of Osborne Compuer Corporation*, which he co-authored with journalist John C. Dvorak, a future *Computer Chronicles* guest.
- Notably, the Osborne and Vector computers were all based on Gary Kildall's CP/M operating system, which may explain the relative lack of any Microsoft or DOS talk in this episode.
- Osborne's use of the phrase "Clash of the Titans" was possibly a reference to the 1981 film starring Laurence Olivier.
- If you watch this episode you'll notice that Lore Harp McGovern speaks English with a pronounced accent. She's originally from Poland.

# Chapter 24: Edward Feigenbaum

For this episode, I’m going to handle things a bit differently. There was only a single guest–Stanford University computer science professor Edward Feigenbaum–and the subject is one that, quite frankly, does not strike me as all that interesting. So rather than do an extended point-by-point recap of the episode, I’m just going to summarize in broad strokes. Trust me, if you had watched the episode, you’d thank me.

## Stanford Professor Discusses AI, Future of Japanese Computing Initiative

You may recall that Feigenbaum appeared in an earlier episode, which I recapped in Part 20. That clip was actually taken from the start of his interview from this episode. And really, this episode covers a lot of the same ground, which focused on knowledge-based or “expert” computer systems. Here, Feigenbaum talked about the state of such systems and the potential threat from Japan’s Fifth Generation computer project.

The Fifth Generation was the term used by Feigenbaum and others to describe to an initiative by the Japanese government to “develop computers with reasoning capabilities, rather than the ability to merely perform calculations,” according to a 1992 article in *The New York Times*. Japan’s Ministry of International Trade and Industry started the 10-year Fifth Generation program in 1982 to basically spur breakthroughs in artificial intelligence.

Speaking with Stewart Cheifet and Gary Kildall–as well as Herbert Lechner, who was along for the ride–Feigenbaum explained that work in the United States on “expert systems” had been going on since about 1965, the beginning of the second decade in AI research. At that time, the emphasis was on “generality,” or the ability of a program to shift its attention from one area to another easily, rather than speed and performance. But the more current research done by people at Stanford focused on creating expert programs, which required encoding a great deal of knowledge about a field into the machine.

Feigenbaum said now–in late 1983–we were starting to see the industrialization of AI, similar to the transition that took place with computing itself during the late 1940s and 1950s, when the technology moved out of the laboratory and into practical uses. For example, he said the University of Pittsburgh now had a "spectacular program" used for clinical diagnosis that knew about 500 different diseases and 3,500 different signs and symptoms.

After a lengthy demonstration of a program developed by Feigenbaum to help the French national oil company Elf Aquitaine identify drilling locations, the conversation eventually shifted to Fifth Generation computers. Feigenbaum co-authored a book on this subject with Pamela McCorduck called, appropriately, *The Fifth Generation–Artificial Intelligence and Japan's Computer Challenge to the World*. He said the Japanese strategy was interesting in that they currently lacked market share in the worldwide computer market. It was difficult to gain market share due to IBM's dominance. So what you had to do was gain a position in an area where IBM has not been active–and that was artificial intelligence. The Japanese goal was therefore to get there first and fast so that they could get a large chunk of the AI/expert systems market before IBM woke up.

Right now, Feigenbaum said, the United States was clearly ahead of the Japanese when it came to computers. But he thought with this new dedicated effort via the Fifth Generation project, the U.S. computer manufacturers had to move pretty fast if they wanted to stay ahead.

## Coleco Launches Adam, Apple Wins Lawsuit Against Clonemaker

Stewart Cheifet presented this week's "Random Access" segment, which I'm dating at sometime during October 1983.

- Continuing our "United States vs. Japan" theme, Cheifet said there was apparently a major effort underway by American computer chip manufacturers to regain their lead over their Japanese counterparts. The U.S. government was working with a newly formed trade group, the Semiconductor Research Corporation, which included 20 major companies that planned to produce the world's first 4 MB computer chip by 1988. Cheifet noted that America used to control the 16 KB market but the Japanese now dominated the 64 KB market.

- That said, the "boom times" were back for chip manufacturing. Cheifet said that U.S. companies were on track to sell $11 billion worth of semiconductors in 1983–an increase of 16 percent over 1982.
- The board of directors at Fortune Systems Corporation, a Silicon Valley computer manufacturer, ousted CEO and founder Gary Friedman, after the company suffered a $3 million loss in the previous quarter and its stock price dropped from a high of $22 per share to a current low of $8.50.
- Telephone company MCI was planning to challenge the U.S. Postal Service with its new MCI Mail service for personal computers. MCI said the system would interface with the Post Office and various courier services so that a user could send electronic mail to people who did not own computers.
- Coleco showed off a production version of its Adam computer to the Boston Computer Society. Cheifet noted that early reports described the Adam as a "not too sophisticated" word processor, but Coleco hinted there would be an additional utility pack.
- Cheifet said that people who owned portable computers and wanted to use them while flying might want to check with their airline first. Many computer users were being told they could not use such devices as there were concerns about radio interference with the navigation systems of airplanes. The *Los Angeles Times* reported that TWA and Republic Airlines were okay with passengers using portable computers, but they were banned on United, PSA, Eastern, Continental, and Western Airlines had banned their use.
- Atari founder Nolan Bushnell, who sold the company to Warner Communications in 1976, was planning a comeback following the expiration of his seven-year non-compete agreement. Cheifet said the rumors were that Bushnell would soon be coming out with "twin laserdisc" video games and he was working on robots to serve pizza at his Pizza Time Theatre chain of restaurants.

- The U.S. Third Circuit Court of Appeals in Philadelphia announced a decision a few weeks ago in Apple's lawsuit against Franklin Computers. Franklin produced the ACE 100, an Apple II-compatible personal computer. Apple alleged the ACE 100 infringed upon its copyrighted operating system. The Third Circuit agreed with Apple's position that a computer program contained on a silicon chip was copyrightable material.
- *Lotus 1-2-3* was the top-selling business software program. Cheifet said during the first six months of 1983, Lotus Corporation sold more than $12 million worth of its popular spreadsheet program. And success was breeding competition: At least 30 software companies announced plans for their own "improved" versions of 1-2-3.
- Finally, Cheifet said, some advertising companies were now thinking about placing ads on software disks or computer networks such as the Dow Jones information service. The early ads would likely be text only and promote software, computer magazines, and financial services.

## The Five Generations of Computers

Let's get the obvious question out of the way–what did "Fifth Generation" mean? Or put another way, what were the first four generations. The simple explanation is as follows:

- First-generation computers (mid-1940) had vacuum tubes.
- Second-generation computers (mid-1950s) had transistors.
- Third-generation computers (mid-1960s) had integrated circuits, i.e., silicon chips with multiple transistors.
- Fourth-generation computers (mid-1970s) were basically the early personal computers.

The term "Fifth Generation" is now used to refer to essentially any computer developed after 1980. In terms of the Japanese Fifth Generation project, it was, as described by the aforementioned *New York Times* article, a 10-year project that largely failed to yield significant commercial results. Indeed, Edward Feigenbaum told the *Times* that while the Japanese government’s investment had yielded some interesting computer designs and software, in the end “no one is using the technology.”

And contrary to the fears raised by Feigenbaum and others that the Fifth Generation project represented an existential threat by the Japanese to U.S. computer dominance, the *Times* noted that “most American computer scientists and executives have long since stopped paying attention to the Fifth Generation project.” The *Times* added that American computer scientists–presumably including Feigenbaum–had overstated the Japanese threat “[i]n order to coax more support from the United States Government for computer science research.”

## Feigenbaum Joined the Air Force (in His Late 50s!)

A decade after his *Computer Chronicles* appearance, Edward Feigenbaum served a three-year stint as chief scientist of the United States Air Force during the Clinton administration. It was no doubt a highlight in a career that began when he was named one of the original members of Stanford University’s computer science faculty in 1965. At Stanford, Feigenbaum developed (with Joshua Lederberg) one of the first expert systems projects called DENDRAL. He would go on to serve as chairman of Stanford’s computer science department and director of its Computer Center. He also co-founded three AI-related startups, including IntelliCorp, which was featured in a previously reviewed *Computer Chronicles* episode.

Feigenbaum retired from the Stanford faculty in 2000. He turned 85 this past January.

## Notes from the Random Access File

- This episode is available at the Internet Archive.
- Ironically, IBM today is primarily known today for its work in AI systems like Watson, a turnaround from its relative lack of interest back in 1983 at the height of the company’s personal computer and mainframe dominance.

- I’d never heard of Fortune Systems before. Its signature product was the Fortune 32:16, described by Old-Computers.com as “the first integrated Unix-based system,” which was released in 1982 and cost $9,000 for a mullti-user configuration including a 10 MB hard drive and 512 KB of memory.
- As for the Franklin ACE 100, I actually had heard of that machine before, thanks to this YouTube video by Adrian Black.
- The Coleco Adam did not fare well on the market, lasting only two years before it was discontinued in 1985.
- If Nolan Bushnell’s “Pizza Time Theatre” featuring robots and arcade games sounds awfully familiar–yup, that’s Chuck E. Cheese. Bushenll opened the first Pizza Time Theatre in San Jose in 1977 when he was still at Atari. Bushnell later bought back the rights to the concept from Warner Communications and proceeded to franchise Chuck E. Cheese. But a few months after this *Chronicles* episode aired, Bushnell was forced to file for Chapter 11 and sold the company to his former business partner.
- As we learned in the last post, Lotus did not take too kindly to all those 1-2-3 clones.
- During Stewart Cheifet’s “Random Access” item on semiconductor sales, they actually displayed a graphic that said “ELEVEN BILLION DOLLARS.”
- Again, I apologize for not doing my usual full-length recap of the episode. But it’s difficult to recap what was basically just a long monologue by Edward Feigenbaum. He just kept talking in one long incredibly unbroken sentence moving from topic to topic so that no one had a chance to interrupt, it was really quite hypnotic…

# Chapter 25: The Lawrence Livermore National Laboratory and Japan’s National Supercomputer Project

You often hear people describe modern smartphones as a “supercomputer in your pocket.” There’s definitely some truth to that, especially when you compare today’s phones with the supercomputers of 40 years ago. In the mid-1980s, supercomputer manufacturers were still struggling with concepts like parallel processing, i.e., breaking down a program into smaller tasks that could run simultaneously on multiple microprocessors. Today, in contrast, multi-core, multi-threaded CPU cores are the norm on just about every personal computing device.

## From the Abacus to the Cray X-MP

Stewart Cheifet opened this *Computer Chronicles* by showing Gary Kildall one of the earliest known computing devices–an abacus. Cheifet joked that if a person was really fast, they could do maybe 2 or 3 operations per second on this 5,000-year-old machine. Today, of course, the subject was supercomputers, which could perform hundreds of millions of operations per second. Why did someone need that kind of incredible speed in a computer? Kildall quipped that for one thing, such incredible speeds helped sell computers. But realistically, there were many problems that did require these kinds of high-performance computers to solve. One example was real-time graphics, where the foreground and background images had to change at a fixed rate, say 30 frames per second. Such graphics required a higher-performance processor to keep up with that frame rate. Other examples Kildall identified were weather modeling and information analysis required by expert systems.

Over this week’s B-roll, Cheifet explained that modern supercomputers were over a million times faster than the serial processing machines of the 1950s. Today’s supercomputers relied on massive parallel architectures, multiple functional units, and comprehensive pipelining to reach processing speeds in excess of 100 million operations per second. The applications for these supercomputers were as distinctly varied as they were specialized, from satellite imagery to super-real graphics, from space exploration to “Doomsday” weapons research.

Cheifet explained that at the Lawrence Livermore National Laboratory (LLNL), the world’s largest user of supercomputers, magnetic fusion experiments required massive computing power made available through the lab’s Octopus Network. This multi-user system shared seven supercomputers, which rented processing time to over 150 U.S. and foreign-based users who gained access through satellite connections.

The fastest of LLNL’s supercomputers, Cheifet said, was the Cray X-MP. This dual-processor supercomputer was reserved for the most secretive and costly activity at LLNL–nuclear weapons design and analysis.

Returning to our favorite theme for this season, Cheifet cautioned that with the Japanese now challenging American dominance in the supercomputer market, research was taking a different approach. Japan produced machines that were not only faster, but which also offered simpler, more portable software that used standard programs aimed at a much wider market.

## Making Algorithms Fit Computers–or Making Computers Fit Algorithms?

In the studio, George A. Michael of LLNL and John Repp from supercomputer manufacturer Control Data Corporation (CDC) joined Kildall and Cheifet. Kildall noted that he had sold one of the first CP/M microcomputers to LLNL about 10 years earlier, which the lab used to operate a microfiche storage peripheral attached to the Octopus Network. So what was going on today at LLNL with supercomputers? Michael explained that the business of the lab was essentially solving partial differential equations, which was then applied to technical subjects like nuclear energy and weapons. This required enormous computational power. In that sense, the supercomputers were not an end in and of themselves, but rather the thing that was used to support the lab's mission.

Kildall asked if LLNL's approach was to buy more computers as their needs grew, or did they try and apply different techniques to use their existing computers more effectively. Michael said they did both. He noted the secret to a supercomputer–or any computer, really–was that there also had to be a language to express ideas. It was also a matter of developing the right algorithms. This required substantial research, and as a result you could not just buy a new computer and expect to immediately produce results 20 times faster. You had to build the thing.

Kildall noted that LLNL worked a lot with CDC. What were the current limitations of its supercomputer technology? Michael said that if you plotted the historic performance of supercomputers, you would see that there was a 100-factor improvement in performance between 1950 and 1960, and again from 1960 to 1970. But from 1970 to 1980 the performance didn't even improve by a factor of 10. Why? Michael said it was partly because components were tired or couldn't go any faster, partly it was that they were doing things incorrectly, and partly it was that the machines required very strange equations and algorithms to really perform.

Kildall pointed out that the CDC STAR-100 was the supercomputer of 10 years ago and it involved a different architecture then today's machines. The STAR was basically an array-processing machine that required different kinds of compilers and software techniques. Kildall asked if that problem had been solved over the past 10 years? Repp replied that when the STAR was first introduced CDC, with the help of LLNL, had to develop a new logic–a new philosophy, as it were–to produce higher performance. The architecture itself had evolved over the past decade largely due to the federal government's investment in research and development. Now, Repp said we were starting to see a supercomputer revolution in the commercial world. The technology was now at the point where companies were jumping on the supercomputer bandwagon.

Cheifet asked what those commercial applications were. Repp said the oil industry was investing in supercomputers in a big way, albeit not totally knowing what they were doing. The main use of supercomputers here was to help search for oil and perform various simulations. Repp said the automobile industry was also getting into supercomputers. The computers allowed manufacturers to crash a car via a simulation as opposed to smashing actual vehicles to conduct safety tests. Supercomputers also helped with vehicle design and productivity, i.e., determining how many cars a company needed to make to turn a profit.

Kildall noted that the famous U.S. Navy computer scientist Captain Grace Hopper used to hand out pieces of wire at lectures to demonstrate how far an electrical signal could travel in a nanosecond (i.e., one-billionth of one second). Hopper claimed the nanosecond was a fundamental limitation on the speed of computing. Was this still true? Michael said he didn't think there was any effective physical limit to computing speed. He said if you told researchers such a limit exists, they would figure out a way around it. Michael said parallel computing was a simple way of getting around the limit Hopper cited. But it required a lot of rethinking in how programs were designed.

Cheifet asked about the technical problems that led Michael to state that increases in supercomputer speeds had hit a wall during this decade. What were the actual barriers? Michael said if you took the STAR as an example, it only performed at blazingly high speeds when its vector unit–which allowed the machine to operate on an array of data simultaneously–was running. But many of the scientific computations required the machine to be kept in the more traditional scalar mode, where the computer could only operate on one piece of data at a time.

So if the scalar mode was slow, Michael said, that slowed down the entire computation process. Michael described it as a "traumatic experience" to convert certain key physics equations, such as those related to the flow of fluid or the migration of neutrons in a nuclear reactor, to work on a vector unit. The lab had to invent special compilers. During this process they learned that it was easy to convert 60% of the program, but it was that remaining 40% that dominated in terms of how the user perceived the speed of the supercomputer.

Kildall restated this to say the real problem was rethinking how to state and present the algorithms to the computer, as opposed to changing the technology itself. Michael said there was some importance in improving the technology, but he essentially agreed with Kildall's assessment. Michael said the key was learning how to remap algorithms onto new architectures. And that required challenging the computer manufacturers in a sense–instead of warping our algorithms to fit your computers, why don't you warp your computers to fit my algorithms? Michael noted this dialog was ongoing, not only with the manufacturers like CDC, but also universities that relied on supercomputers. He expected this would lead to some breakthroughs by the end of the 1980s.

Cheifet asked Repp for his response. Repp said he didn't disagree with Michael's analysis. But if you looked at the commercial world, instead of rewriting code they were relying on conversions instead. It was time consuming and expensive to write entirely new code. And many commercial operations had code that had been written and built up over the years for scalar-based machines, and it would require an enormous effort to rewrite all those programs for vector-based computers.

## Japan Adopts a Public-Private Approach to Challenge U.S. Supercomputer Dominance

The final segment was largely outside the studio and consisted of a series of interviews Cheifet conducted in Japan with that country's leading supercomputer scientists. In an opening narration, Cheifet said that at the University of Tokyo, the computer science department recently acquired a HITAC S-810, one of the fastest supercomputers in the world, which was capable of performing 800 million floating-point operations per second (FLOPS). Hitachi, the manufacturer of the S-810, was one of ten companies participating in Japan's National Supercomputer Project, a collaboration between the country's public and private sectors. The project's goal was to eventually produce a supercomputer that was 10 times faster than today's machines–one that could perform 10 billion FLOPS.

Cheifet interviewed Dr. Kashiwagi Hiroshi, the leader of the supercomputer project. Speaking through an interpreter, Kashiwagi explained that this expected increase in speed would be accomplished through chip design. He noted the architecture itself would have to be 100 times faster than present supercomputers in order achieve a tenfold increase in speed. To accomplish this, the project was exploring the use of Josephson junctions–a "sandwich" made with two layers of superconducting material around a layer of non-conductive material–as well as high-electron mobility transistors, and Gallium arsenide, the latter of which was a seen as a potentially superior substitute for silicon in producing computer chips. Kashiwagi added that the new architecture itself would require development of a much higher degree of parallel processing than was currently available.

In narration, Cheifet said that testing of these new materials and designs were already underway at the Japanese government's Electrotechnical Library, located in a part of Tsukuba, Japan, known as "Science City." Because of the extreme heat created by this new high-speed circuitry, Cheifet said the chips had to be cooled to extreme temperatures–around -270 degrees centigrade.

Cheifet reiterated that higher computational speeds in supercomputers would also depend on developing new ways of handling data through parallel processing and pipeline architecture. Kashiwagi said the best way to explain these concepts was to think of a conveyor belt assembly line in an automobile factory. The conveyor belt started with one part and picked up additional components along the way until the car was completed. This was an efficient way to make one product using one line. Now, if you increased the number of conveyor belts you could produce more cars. But if you increased the number of belts by 100, you shouldn't expect to increase the output by 100 times, because it was difficult to properly allocate the work to 100 different assembly lines simultaneously. You needed a good manager to control the process. Similarly, you shouldn't expect 100 times the output from a supercomputer with 100 times as many processors, although you would still see a substantial increase in processing capacity.

Cheifet next interviewed Dr. Aiso Hideo, a professor of electrical engineering at Keio University, who cautioned that the Josephson junction could only be used in a very low temperature environment. If you moved the components from its lowest temperature to room temperature, the junctions would break down. He therefore thought it would be very difficult to extend such parts to a large-scale computer.

Kashiwagi added that another problem with developing these new machines was the need for new languages. Most research centers currently relied on FORTRAN. But the success of the Japanese project's machines depended on whether users were willing to change to a different language. He noted that while there were some new languages in development, very few were currently in widespread use, which posed a potentially big problem.

In a final interview, Dr. Mizuno Yukio, a senior vice president with Japanese computer manufacturer NEC, explained some of the "unusual arrangements" between the government and the private sector to make the supercomputer project work. Specifically, Mizuno said any patents arising from the technology developed by the project belonged to the government. But the participating companies had the right to use those patents and develop their own products accordingly. In that sense, the Japanese project was still promoting free competition.

Indeed, Kashiwagi pointed out that good technology was always born out of a competitive environment. If the United States were the only country developing supercomputers, we wouldn't expect the kind of breakthroughs that would come from competition. Similarly, if Japan monopolized the research, you would not see the current level of progress continue. Kashiwagi said he therefore hoped that Japan would always be a good competitor with the United States.

## Could Japan Deliver on Its Promises?

Returning to the studio, Cheifet asked George Michael if there really was a significant difference in the Japanese and American approach to supercomputers. Michael said there was a difference, but he didn't think that difference explained the current success of the Japanese companies in the market. He cited Fujitsu as an example. In discussions with LLNL, Michael said Fujitsu managed to write a new vector-based compiler for approximately 5,000 existing FORTRAN programs. Michael said it was the “best looking compiler” they’d ever seen, and it suggested the next generation of Japanese supercomputers would pose a serious threat to the American manufacturers.

Asked for CDC’s response, John Repp said the Japanese technology was unproven. The Japanese machines were still in the research environment and had yet to be delivered to any customers. And even if the Japanese started delivering machines by 1986, he said that American companies would already be out with supercomputers that were 10 times more powerful than current models. Extrapolating further, Repp said we were looking at machines improved by an order of 100 by 1990. So he felt that while the Japanese were tough competitors, America would respond with its own excellent technology.

Cheifet asked if there were any bigger consequences to this USA-Japan competition, say with respect to national defense. Michael said we weren’t faced with a life-or-death situation. But he did feel that if the U.S. companies didn’t get busy then the Japanese supercomputers would become dominant in terms of speed and price. He noted that aside from Cray and CDC, there really weren’t any other major players in the domestic supercomputer market. And that might not be enough to meet the “structured challenge” posed by the Japanese supercomputer project.

## CDC Pioneered Supercomputer Market, But Later Collapsed Under Its Own Weight

So if you’re wondering who ultimately won the battle for supercomputer dominance–the United States or Japan–the answer would appear to be China. According to Statista.com, three Chinese companies–Lenovo, Sugon, and Inspur–sold over 60 percent of the world’s top 500 supercomputers as of June 2020. Hewlett-Packard Enterprise, which now includes Cray, is the largest American vendor on that list with roughly 15 percent of the market. Fujitsu is the largest Japanese manufacturer on the list with a market share of less than 3 percent.

One name you will not see on this list is Control Data Corporation. CDC broke apart in the late 1980s after more than two decades of pioneering work in the supercomputer market. A group of employees at a St. Paul, Minnesota, computer company called Engineering Research Associates left to start CDC in 1957. Donald M. Hall offered this account of CDC's founding in a 2017 retrospective for the *Minneapolis Star-Tribiune*:

> [T]he initial employee group began to raise money by calling on friends, relatives and strangers in presentations over lunch, at house parties and in any other possible way. By September 1957, they had received commitments for $1.2 million, double the amount anticipated in the prospectus. The state securities commissioner was dumbfounded. As he said, there was no product, no plant, no customers, no money and almost no employees. When the subscription was completed, he received calls from around the country asking: "Is this true?" It was, he told them, as he had approved the new company's offering document, only eight pages long, and found no legal objection.
>
> The company's name was chosen from two columns of words by matching combinations that sounded appropriate. No one was much satisfied with the final selection, but they accepted it with resignation, not knowing, of course, that it would quickly electrify the Twin Cities local stock market and within six years be listed on the New York Stock Exchange. From its initial price of $1 per share in 1957, the stock in 1961 traded at $100 a share.

CDC helped define the supercomputer market in the 1960s and 1970s with machines like the CDC 6600, released in 1964, and the aforementioned CDC STAR-100. During this period, CDC's lead designer was Seymour Cray, who left to start his own firm, Cray Research, in 1972. Meanwhile, throughout the 1970s CDC expanded into a number of other business lines, including commercial credit, computer peripherals like disk drives, and even ticket sales.

By the 1980s, the whole thing had apparently become too unwieldy. An account provided by Minnesota Computing History suggested that CDC "failed to transition quickly to the new realities brought on by personal computers” during the early 1980s. After CDC posted a loss of $567 million in 1985, it tried to reorganize, which in reality meant slowly selling off parts of the company. In 1992, CDC split itself into two companies: Ceridian, which contained the remaining services components of the business; and Control Data Systems (CDS), which was the remainder of the computer hardware business.

Ceridian is still in business today as the publicly traded Ceridian HCM, Inc., a provider of human resources software. BT Group (formerly British Telecom) acquired what was left of CDS in 1999 and merged it with BT’s Syntegra subsidiary.

## Repp Rose from HR Assistant to VP of Sales

John Repp had a 22-year career with CDC. According to a May 2020 profile by Nancy Vogt in the *Pine and Lakes Echo Journal*, Repp left the Navy after serving as a lieutenant and went looking for a management job. When he couldn’t find one–due to his poor college GPA–Repp ended up as a personnel assistant at CDC. He later managed CDC’s data centers before shifting to sales, which Repp told Vogt was his “true passion.”

Repp left CDC and joined Alan Shugart’s Seagate Technology in 1989 as its vice president of sales. He retired from Seagate in 1995. According to Vogt, Repp is now enjoying his retirement with his wife of 60 years, three daughters, and seven grandchildren.

## George A. Michael (1926 - 2008)

George Michael worked at the Lawrence Livermore National Laboratory for 38 years. Initially hired as a theoretical physicist in 1953, Michael would go on to play a major role in the laboratory’s supercomputer program. Even after his formal retirement in 1991, he continued to work part-time documenting the history of the laboratory on his personal website, which remains active today in partially finished form.

Michael detailed LLNL’s history of supercomputer acquisitions between 1953 to 1982. With respect to the CDC STAR-100, he commented:

> The STAR 100 was the first vector machine in design, but because of its other complexities, was not the first to enter the marketplace. Vector computing was then rather new, and difficult in adaptation. A major problem on the STAR was the disparity in speeds between vector and scalar computations. The STAR provided the largest memories in use at the Lab: one or two million words. They were used as time sharing servers, but it was very hard to do that and efficiency suffered. They were late in delivery. It was difficult to produce efficient programs, and generally failed to make an impression on the marketplace. One of the designers at the Lab remarked that his team had spent seven years to gain mastery of the STAR, and that was seven years of not improving the physics, and he was never going to do that again.

In addition to his work at the laboratory, Michael also co-founded the International Conference for High Performance Computing, Networking, Storage and Analysis, more commonly known today as “SC,” in 1988. Michael died in June 2008 at the age of 82.

## Notes from the Random Access File

- This episode is available at the Internet Archive. I would place the date towards the end of 1983, based largely on Gary Kildall referring to “Captain Grace Hopper.” President Reagan promoted Hopper to Commodore–or Rear Admiral (lower half) in modern parlance–in December 1983 after Congress passed a joint resolution requesting the appointment.
- LLNL is not named after Lawrence Livermore. The lab is based in Livermore, California. The “Lawrence” was added in honor of Ernest Lawrence, the co-founder of the facility, which was originally called the University of California Radiation Laboratory at Livermore when it started in 1952.
- LLNL is still very much active today. It is part of the U.S. Department of Energy and has an annual operating budget of around $2.33 billion.

- The Josephson junction is named for Brian Josephson, who shared the 1973 Nobel Prize in Physics with two others for his work leading to its creation.
- The IPSJ Computer Museum has some good background on two of the Japanese interview subjects from this episode, Dr. Aiso Hideo and Dr. Mizuno Yukio. Mizuno passed away in 2003.

# Appendix

In my posts covering the first season of *Computer Chronicles*, I noted that some of the episode recordings included interstitial segments presented by Herbert Lechner, who also occasionally guest hosted for Gary Kildall during this 1983-84 period. These segments were part of a "telecourse" marketed by Wadsworth Publishing Company as part of its "continuing education professional series." The idea was to pair the first 26 *Chronicles* episodes with a companion textbook for students to follow along and learn more in-depth about the topics discussed on the show. Lechner authored the book (as H.D. Lechner) in addition to hosting the special segments.

The book is difficult to find in print form these days. I saw a copy offered for sale on eBay for $94, and some academic libraries report having copies on their shelves. But fortunately, you can access the full text of the book at the Internet Archive. Well, actually you can "rent" access one hour at a time via the Archive.

I've had a chance to look through *The Computer Chronicles* in book form. The text is divided into five sections–Computer Hardware, Computer Software, Computer Applications, Computers and People, and Computers in the Future–encompassing a total of 26 chapters. Each chapter corresponds to an episode of the television series. However, the order is significantly different. For example, while the second episode broadcast was on "Integrated Software," that is chapter 10 in the book. So if you were following along with the show via the telecourse and the textbook, you would be watching everything out of order. (Presumably, the telecourse episodes did not include the time-sensitive "Random Access" segments, which would have only confused matters further.)

It's also clear the book was written at the same time as the first batch of episodes were in production. Some chapters do include pull-quotes from guests on the actual episodes. But for the most part, you don't really need to have watched the show to read and understand the text. The book also had a publication date of March 1984, which was shortly after the release of the

Macintosh, which was only referred to in a picture, while some of the text still referred to the Apple Lisa as an active product.

## The First “Lost” Episode

I’m not going to review the chapters in any great detail, as they basically just provide additional technical and historical background for the subjects discussed in the televised episodes. But I will briefly point to chapter 8 of the book, entitled “Application Development.” This chapter was meant to correspond with the 20th broadcast episode on “Software Utilities,” which I have not covered on the blog.

The reason for my lack of coverage is quite simple: There’s no publicly available recording of the episode–at least not yet. There are, in fact, several episodes presently lost or missing from the Internet Archive’s repository, which was taken from original tapes provided by Stewart Cheifet. I’ve been told by reliable sources that the Archive does now have a complete set of tapes, so eventually these lost episodes may once again see the light of day. But for now, the “Software Utilities” episode remains something of a mystery.

The chapter in the book also provided me with no useful clues as to who appeared on this episode or what products were discussed. Given the late place in the production order, I suspect Lechner wrote this chapter before the episode actually taped. His text focused on explaining the process of software development and software engineering, but he only spoke in generalities without delving into particular software products.

With respect to the state of software engineering in 1983-84, Lechner had this to say:

> Several major issues affect application development productivity, not the least of which are the fact that it is poorly defined and hard to quantify, and that little valid data has been collected on which to base productivity analyses. Nonetheless, a number of factors illustrate the importance of software engineering. First, more effort goes into maintaining applications than in developing them (43 percent of the

> costs over an application life are generally for development, 49 percent are for maintenance, and 8 percent are for other reasons). Second, application development techniques have high initial costs in hardware and training. Third, university and college education in the area of large-scale application development is weak. Large-system development can take many years to complete, and it is difficult to have "laboratory training" in this process at school. Fourth, methods by which software engineers can be motivated and retained are not well understood. And fifth, users are, as indicated, ambivalent in many cases in their attitudes towards application specification.

Expounding on that fourth point–how to motivate and retain software engineers–Lechner said he thought the best practice was to simply hire better engineers in first place:

> In selecting programmers and analysts, an organization should use validated application programmer testing, hire those with valuable and applicable experience, and adhere to proven employment practices such as good reference checking.

I suspect a lot of people working today as software engineers would find this attitude rigid, patriarchal, and exclusionary. Of course, Lechner's career was spent largely in big business and academia. He started at IBM in the 1950s and later worked for Singer Corporation and American Express before joining SRI International as a vice president. So he wasn't exactly the type of person you'd expect to embrace the 1980s bedroom BASIC coder or the 21st century self-taught Python developer.

## A Norton Footnote

As I said, we don't have any solid data on this missing "Software Utilities" episode. But I believe that one of the guests was Peter Norton, creator of *The Norton Utilities*. There's a few reasons for this. First, a reader alerted me sometime back to an episode featuring Peter Norton, and I couldn't find one on the publicly available archive. Second, Stewart Cheifet himself referred to an episode featuring Norton during a 2013 interview with Leo Laporte. And given the listed title of the episode was "Software Utilities," it makes sense that Norton would have been a guest, as he was still actively developing *Norton Utilities* during this time period.

# About the Author

S.M. Oliva is a freelance writer with over 25 years experience authoring original content for the web. He also spent over a decade working as a paralegal in probate, trusts, and estate planning. If you are looking to hire a freelance blogger or content creator for your own project, please feel free to contact him.